

[MS-CUSTOMUI]: Custom UI XML Markup Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.aspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

| Date | Revision History | Revision Class | Comments |
|------------|------------------|----------------|--|
| 01/15/2009 | 1.0 | Major | Initial Availability |
| 07/13/2009 | 1.01 | Major | Revised and edited the technical content |
| 08/28/2009 | 1.02 | Editorial | Revised and edited the technical content |
| 11/06/2009 | 1.03 | Editorial | Revised and edited the technical content |
| 02/19/2010 | 2.0 | Editorial | Revised and edited the technical content |
| 03/31/2010 | 2.01 | Editorial | Revised and edited the technical content |
| 04/30/2010 | 2.02 | Editorial | Revised and edited the technical content |
| 06/07/2010 | 2.03 | Editorial | Revised and edited the technical content |
| 06/29/2010 | 2.04 | Editorial | Changed language and formatting in the technical content. |
| 07/23/2010 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 09/27/2010 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 11/15/2010 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 12/17/2010 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 03/18/2011 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 06/10/2011 | 2.04 | No change | No changes to the meaning, language, or formatting of the technical content. |

Table of Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 5 |
| 1.1 | Glossary | 5 |
| 1.2 | References..... | 5 |
| 1.2.1 | Normative References..... | 5 |
| 1.2.2 | Informative References | 5 |
| 2 | Custom UI | 6 |
| 2.1 | Parts | 6 |
| 2.1.1 | Quick Access Toolbar Customizations Part | 6 |
| 2.1.2 | Ribbon Extensibility Part..... | 7 |
| 2.2 | Elements | 8 |
| 2.2.1 | box (Box Grouping Container)..... | 8 |
| 2.2.2 | button (Button)..... | 12 |
| 2.2.3 | button (Unsize Button)..... | 23 |
| 2.2.4 | button (Button Inside of a Split Button) | 32 |
| 2.2.5 | buttonGroup (Button Grouping Container) | 41 |
| 2.2.6 | checkBox (Check Box) | 45 |
| 2.2.7 | comboBox (Combo Box)..... | 55 |
| 2.2.8 | command (Repurposed Command)..... | 67 |
| 2.2.9 | commands (List of Repurposed Commands) | 69 |
| 2.2.10 | contextualTabs (List of Contextual Tab Sets)..... | 69 |
| 2.2.11 | control (Unsize Control Clone) | 70 |
| 2.2.12 | control (Control Clone)..... | 78 |
| 2.2.13 | control (Quick Access Toolbar Control Clone) | 88 |
| 2.2.14 | customUI (Custom UI Document Root)..... | 97 |
| 2.2.15 | dialogBoxLauncher (Dialog Box Launcher) | 98 |
| 2.2.16 | documentControls (List of Document-Specific Quick Access Toolbar Controls) | 99 |
| 2.2.17 | dropDown (Drop-down Control)..... | 100 |
| 2.2.18 | dynamicMenu (Unsize Dynamic Menu)..... | 113 |
| 2.2.19 | dynamicMenu (Dynamic Menu)..... | 123 |
| 2.2.20 | editBox (Edit Box) | 134 |
| 2.2.21 | gallery (Gallery) | 144 |
| 2.2.22 | gallery (Unsize Gallery) | 161 |
| 2.2.23 | group (Group)..... | 178 |
| 2.2.24 | item (Selection Item)..... | 186 |
| 2.2.25 | labelControl (Text Label) | 188 |
| 2.2.26 | menu (Unsize Menu) | 197 |
| 2.2.27 | menu (Menu with Title) | 207 |
| 2.2.28 | menu (Menu) | 218 |
| 2.2.29 | menu (Dynamic Menu Root XML Element) | 229 |
| 2.2.30 | menuSeparator (Menu Separator) | 231 |
| 2.2.31 | officeMenu (Office Menu) | 235 |
| 2.2.32 | qat (Quick Access Toolbar) | 236 |
| 2.2.33 | ribbon (Ribbon) | 237 |
| 2.2.34 | separator (Separator) | 238 |
| 2.2.35 | sharedControls (List of Shared Quick Access Toolbar Controls) | 241 |
| 2.2.36 | splitButton (Unsize Split Button) | 242 |
| 2.2.37 | splitButton (Split Button with Title) | 251 |
| 2.2.38 | splitButton (Split Button)..... | 259 |
| 2.2.39 | tab (Tab)..... | 269 |

| | | |
|----------|--|------------|
| 2.2.40 | tabs (List of Tabs) | 273 |
| 2.2.41 | tabSet (Contextual Tab Set) | 274 |
| 2.2.42 | toggleButton (Unsize Toggle Button) | 275 |
| 2.2.43 | toggleButton (Toggle Button) | 285 |
| 2.2.44 | toggleButton (Toggle Button Inside of a Split Button) | 295 |
| 2.3 | Simple Types | 305 |
| 2.3.1 | ST_BoxStyle (Box Style) | 305 |
| 2.3.2 | ST_Delegate (Callback Function Name) | 305 |
| 2.3.3 | ST_GalleryItemWidthHeight (Gallery Item Width or Height) | 308 |
| 2.3.4 | ST_GalleryRowColumnCount (Gallery Row or Column Count) | 308 |
| 2.3.5 | ST_ID (Control ID) | 309 |
| 2.3.6 | ST_ItemSize (Menu Item Size) | 310 |
| 2.3.7 | ST_Keytip (Keytip) | 310 |
| 2.3.8 | ST_LongString (Long String) | 311 |
| 2.3.9 | ST_QID (Qualified Control ID) | 312 |
| 2.3.10 | ST_Size (Control Size) | 313 |
| 2.3.11 | ST_String (Short String) | 314 |
| 2.3.12 | ST_StringLength (String Length) | 315 |
| 2.3.13 | ST_UniqueID (Custom Control ID) | 315 |
| 2.3.14 | ST Uri (Image Relationship ID) | 316 |
| 3 | Appendix A: Custom UI Control ID Tables | 318 |
| 3.1 | idMso Tables | 318 |
| 3.1.1 | Word 2007 | 318 |
| 3.1.2 | Excel 2007 | 376 |
| 3.1.3 | PowerPoint 2007 | 414 |
| 3.2 | imageMso Table | 442 |
| 4 | Appendix B: Product Behavior | 551 |
| 5 | Change Tracking | 552 |
| 6 | Index | 553 |

1 Introduction

In creating an interoperable implementation, it may be helpful to understand specific implementation choices made by other products implementing the same standard. For example, portions of the standard may provide only general guidance, leaving specific implementation choices up to the application implementer; in some circumstances it may be helpful for other implementers to understand those choices.

The information contained in this document provides information about how Microsoft® Office Word 2007, Microsoft® Office Excel® 2007, Microsoft® Office PowerPoint® 2007, Microsoft® Word 2010, Microsoft® Excel® 2010 and Microsoft® PowerPoint® 2010 implement UI customization in the context of ECMA-376 Office Open XML File Formats [\[ECMA-376\]](#).

1.1 Glossary

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[ECMA-376] ECMA International, "Office Open XML File Formats", 1st Edition, ECMA-376, December 2006, <http://www.ecma-international.org/publications/standards/Ecma-376.htm>

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

1.2.2 Informative References

None.

2 Custom UI

The subordinate clauses specify the semantics for the Custom UI XML markup contained within the ECMA-376 Office Open XML File Formats [\[ECMA-376\]](#). These semantics describe customization of the UI interface. Examples in the following clauses highlight customizations in the context of the Microsoft Office Fluent interface (UI) but the concepts extend naturally to any user interface.

Customization of the UI is accomplished via the addition of parts containing Custom UI XML markup to the containing document package.

2.1 Parts

The parts described in the subordinate sections detail the additional part types utilized by CustomUI in an ECMA-376 Office Open XML File Formats [\[ECMA-376\]](#) file.

2.1.1 Quick Access Toolbar Customizations Part

| | |
|----------------------|---|
| Content Type: | application/xml |
| Root Namespace: | http://schemas.microsoft.com/office/2006/01/customui |
| Source Relationship: | http://schemas.microsoft.com/office/2006/relationships/ui/userCustomization |

An instance of this part type contains information about the quick access toolbar customizations specific to the containing package.

[Example: A user customized the quick access toolbar for his WordProcessingML document to contain the UI controls that they commonly use. end example]

A package is permitted to contain at most one Quick Access Toolbar Customizations part, and that part shall be the target of a relationship in the package-relationship item for the document.

[Example: The following package part-relationship item contains a relationship to a Quick Access Toolbar Customizations part, which is stored in the ZIP item /userCustomization/customUI.xml:

```
<Relationships xmlns="...">
  <Relationship Id="rId2"
    Type="http://.../2006/relationships/ui/userCustomization"
    Target="/userCustomization/customUI.xml" />
</Relationships>
```

end example]

The root element for a part of this content type shall be `customUI`.

[Example: The following Quick Access Toolbar Customizations content markup would specify that the control with ID `SpellingAndGrammar` should be added to the quick access toolbar for the package:

```
<mso:customUI xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
  <mso:ribbon>
    <mso:qat>
      <mso:documentControls>
        <mso:control idQ="mso:SpellingAndGrammar" visible="true" />
      </mso:documentControls>
    </mso:qat>
  </mso:ribbon>
</mso:customUI>
```

```

</mso:qat>
</mso:ribbon>
</mso:customUI>

```

end example]

A Quick Access Toolbar Customizations part shall be located within the package containing the source relationship (expressed syntactically, the `TargetMode` attribute of the `Relationship` element shall be `Internal`).

A Quick Access Toolbar Customizations part shall not have implicit or explicit relationships to any other part defined by ECMA-376 Office Open XML File Formats [\[ECMA-376\]](#).

2.1.2 Ribbon Extensibility Part

| | |
|----------------------|--|
| Content Type: | application/xml |
| Root Namespace: | <code>http://schemas.microsoft.com/office/2006/01/customui</code> |
| Source Relationship: | <code>http://schemas.microsoft.com/office/2006/relationships/ui/extensibility</code> |

An instance of this part type contains information about the ribbon customizations specific to the containing package.

[Example: A SpreadsheetML document that represents a timecard could contain custom UI controls to guide the user in filling out the timecard. end example]

A package is permitted to contain at most one Ribbon Extensibility part, and that part shall be the target of a relationship in the package-relationship item for the document.

[Example: The following package part-relationship item contains a relationship to a Ribbon Extensibility part, which is stored in the ZIP item `/customUI/customUI.xml`:

```

<Relationships xmlns="...">
<Relationship Id="rId5"
Type="http://.../2006/relationships/ui/extensibility"
Target="/customUI/customUI.xml" />
</Relationships>

```

end example]

The root element for a part of this content type shall be `customUI`.

[Example: The following Ribbon Extensibility content markup would specify that the ribbon tab with ID `TabHome` should be hidden for the containing package:

```

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon>
<tabs>
<tab idMso="TabHome" visible="false" />
</tabs>
</ribbon>
</customUI>

```

end example]

A Ribbon Extensibility part shall be located within the package containing the source relationship (expressed syntactically, the `TargetMode` attribute of the `Relationship` element shall be `Internal`).

A Ribbon Extensibility part is permitted to have explicit relationships to the following parts defined by ECMA-376 Office Open XML File Formats [\[ECMA-376\]](#):

Image Part ("[\[ECMA-376\]](#) Part 1 section15.2.13; Image Part")

2.2 Elements

A Custom UI document contains customizations of an application's UI. Customizations are mainly of two types:

- Modifications of the application's built-in UI [Example: Hiding or disabling built-in UI controls, repurposing command actions, and so on. end example]
- Creation of custom UI controls [Example: Creating a custom ribbon tab, menu item, or quick access toolbar button. end example]

[Example: Consider the following Custom UI document:

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <commands>
    <command idMso="Bold" enabled="false" />
  </commands>
  <ribbon>
    <tabs>
      <tab idMso="TabHome" visible="false" />
      <tab id="CustomTab" label="Custom Tab">
        <group id="CustomGroup" label="Custom Group">
          <button id="CustomButton" label="Custom Button"
            size="large" imageMso="HappyFace" onAction="OnButtonClick" />
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

This example disables the command with an ID of `Bold`, hides the ribbon tab with an ID of `TabHome` and creates a new custom ribbon tab with a custom button in it. end example]

2.2.1 box (Box Grouping Container)

This element specifies a grouping container control that can be used to align controls vertically or horizontally. `Box` elements can be nested to create complex UI layouts.

[Example: Consider a group of controls which are laid out horizontally, as follows:

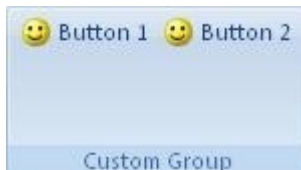


Figure 1: Controls grouped horizontally

This layout would be specified using the following XML fragment:

```
<box id="box" boxStyle="horizontal">
  <button id="button1" label="Button 1" imageMso="HappyFace" />
  <button id="button2" label="Button 2" imageMso="HappyFace" />
</box>
```

This is contrasted to the default vertical layout which would be used if the `box` element is not specified. end example]

| Parent Elements |
|---|
| <code>box</code> (§2.2.1); <code>group</code> (§2.2.23) |

| Child Elements | Subclause |
|--|-----------|
| <code>box</code> (Box Grouping Container) | §2.2.1 |
| <code>button</code> (Button) | §2.2.2 |
| <code>buttonGroup</code> (Button Grouping Container) | §2.2.5 |
| <code>checkBox</code> (Check Box) | §2.2.6 |
| <code>comboBox</code> (Combo Box) | §2.2.7 |
| <code>control</code> (Control Clone) | §2.2.12 |
| <code>dropDown</code> (Drop-down Control) | §2.2.17 |
| <code>dynamicMenu</code> (Dynamic Menu) | §2.2.19 |
| <code>editBox</code> (Edit Box) | §2.2.20 |
| <code>gallery</code> (Gallery) | §2.2.21 |
| <code>labelControl</code> (Text Label) | §2.2.25 |
| <code>menu</code> (Menu) | §2.2.28 |
| <code>splitButton</code> (Split Button) | §2.2.38 |
| <code>toggleButton</code> (Toggle Button) | §2.2.43 |

| Attributes | Description |
|-----------------------------------|--|
| <code>boxStyle</code> (Box Style) | Specifies the layout direction for the child controls inside of the <code>box</code> element. If this attribute is omitted, then the child controls should be laid out horizontally. [Example: Consider a group of controls which shall be laid out vertically. This is specified using the following XML: |

| Attributes | Description |
|----------------------------------|--|
| | <pre><box id="box" boxStyle="vertical"> ... </box></pre> <p>▪ end example]</p> <p>The possible values for this attribute are defined by the ST_BoxStyle simple type (§2.3.1).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the IsButtonVisible callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of MyButton. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon></pre> |

| Attributes | Description |
|---|---|
| | <p><code></customUI></code></p> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before</p> |

| Attributes | Description |
|--|--|
| | <p>the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <ul style="list-style-type: none"> The possible values for this attribute are defined by the XML Schema boolean datatype. |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Box">
  <group ref="EG_Controls" minOccurs="0" maxOccurs="1000"/>
  <attributeGroup ref="AG_IDCustom"/>
  <attributeGroup ref="AG_Visible"/>
  <attributeGroup ref="AG_PositionAttributes"/>
  <attribute name="boxStyle" type="ST_BoxStyle" use="optional"/>
</complexType>
```

2.2.2 button (Button)

This element specifies a standard push-button control that performs an action when clicked.

[Example: Consider a button control, as follows:



Figure 2: A button control

This would be specified using the following XML fragment:

```
<button id="button" label="Button" imageMso="HappyFace" />
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Attributes | Description |
|---------------------------|---|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="483 982 1052 1087" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> |


| Attributes | Description |
|--|---|
| | The possible values for this attribute are defined by the XML Schema boolean datatype. |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |


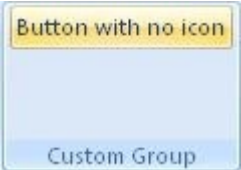
| Attributes | Description |
|---|---|
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when</p> |


| Attributes | Description |
|---|--|
| | <p>the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getSize (getSize Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 506 1101 531"><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getSupertip (getSupertip Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 867 1203 892"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getVisible (getVisible Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1228 1166 1253"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>id (Control ID)</p> | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1617 1015 1642"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type</p> |


| Attributes | Description |
|-----------------------------|--|
| | (§2.3.13). |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |

| Attributes | Description |
|--|---|
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|--|
| | <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>onAction</code> (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> |

| Attributes | Description |
|--|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1308 275" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" label="Button" onAction="ButtonClicked" /> </pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 989 1218 1041" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /> </pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>showImage</code> (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 1598 987 1650" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" showImage="false" label="Button with no icon" /> </pre> <p>end example]</p> |

| Attributes | Description |
|------------------------|--|
| | The possible values for this attribute are defined by the XML Schema boolean datatype. |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 485 1166 533"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| size (Control Size) | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 1062 1052 1083"><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p> |

| Attributes | Description |
|------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Button">
  <complexContent>
    <extension base="CT_ButtonRegular">
```

```

<attributeGroup ref="AG_SizeAttributes"/>
</extension>
</complexContent>
</complexType>

```

2.2.3 button (Unsize Button)

This element specifies a push-button which, due to its location, cannot have its size changed. The `size` attribute is not present. This element otherwise behaves like the regular `button` (§2.2.2) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5) ; dialogBoxLauncher (§2.2.15) ; documentControls (§2.2.16) ; dropDown (§2.2.17) ; gallery (§2.2.21) ; gallery (§2.2.22) ; menu (§2.2.28) ; menu (§2.2.26) ; menu (§2.2.29) ; menu (§2.2.27) ; officeMenu (§2.2.31) ; sharedControls (§2.2.35) |

| Attributes | Description |
|-------------------------------------|--|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="482 934 1047 1037" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_LongString</code> simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |


| Attributes | Description |
|--|---|
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel | Specifies the name of a callback function which should be called to determine the |



| Attributes | Description |
|--|--|
| Callback) | <p>label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 344 1127 369"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 732 1230 758"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowImage (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1155 1256 1180"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1604 1256 1629"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end</p> |


| Attributes | Description |
|--|---|
| | <p>example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |

| Attributes | Description |
|-------------------------------------|---|
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case ex is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a tab in that namespace with an ID of OtherTab. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of ForestPic. end example]</p> <p>The possible values for this attribute are defined by the ST_Uri simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of Bold. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |

| Attributes | Description |
|---|---|
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertBeforeMso (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> |

| Attributes | Description |
|-------------------------------------|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1308 331"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 779 1203 800"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1087 1078 1108"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1360 1308 1381"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p> |

| Attributes | Description |
|------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label</p> |

| Attributes | Description |
|--|--|
| | <p>is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p><code>tag</code> (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p><code>visible</code> (Control Visibility)</p> | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end</p> |

| Attributes | Description |
|------------|--|
| | example] The possible values for this attribute are defined by the XML Schema boolean datatype. |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ButtonRegular">
  <complexContent>
    <extension base="CT_Control">
      <attributeGroup ref="AG_Action"/>
      <attributeGroup ref="AG_Enabled"/>
      <attributeGroup ref="AG_Description"/>
      <attributeGroup ref="AG_Image"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.4 button (Button Inside of a Split Button)

This element specifies a push-button that is a child of a split button control. The `visible` and `getVisible` attributes are not present because the visibility is controlled by the split button. This element otherwise behaves like the `unsized button` (§2.2.3) element.

| Parent Elements |
|--|
| <code>splitButton</code> (§2.2.38); <code>splitButton</code> (§2.2.36); <code>splitButton</code> (§2.2.37) |

| Attributes | Description |
|--|---|
| <code>description</code> (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="482 1274 1050 1377" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_LongString</code> simple type (§2.3.8).</p> |
| <code>enabled</code> (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute</p> |


| Attributes | Description |
|--|---|
| | <p>is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> |

| Attributes | Description |
|---|--|
| Callback) | <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> | Specifies the name of a callback function which should be called to determine |


| Attributes | Description |
|--|---|
| (getShowLabel Callback) | <p>whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |

| Attributes | Description |
|-----------------------------|--|
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in | Specifies the ID of a built-in image which should be used as the icon of this control. |

| Attributes | Description |
|---|---|
| Image ID) | <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ...</pre> |

| Attributes | Description |
|---|--|
| | <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>onAction</code> (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|---|
| | <pre data-bbox="516 218 1308 241" style="background-color: #f0f0f0;"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p data-bbox="488 273 1382 323">This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p data-bbox="488 331 1357 382">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 409 358 459"><code>screentip</code> (Screentip)</p> | <p data-bbox="488 409 1276 432">Specifies a string which should be shown as the screentip for this control.</p> <p data-bbox="488 443 1354 520">The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="488 529 1101 552">[Example: Consider a button with a screentip, as follows:</p> <div data-bbox="488 562 980 894" data-label="Image"> <p>The image shows a user interface with a button that has a yellow smiley face icon and the text 'Button'. Below the button is a tooltip with a red border containing the text 'This is the screentip'. The button is part of a 'Custom Group'.</p> </div> <p data-bbox="488 905 1109 928">This would be specified using the following XML fragment:</p> <pre data-bbox="516 955 1216 1005" style="background-color: #f0f0f0;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="488 1037 634 1060">end example]</p> <p data-bbox="488 1071 1325 1121">The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p data-bbox="237 1144 431 1194"><code>showImage</code> (Show Image)</p> | <p data-bbox="488 1144 1053 1167">Specifies whether this control should display an icon.</p> <p data-bbox="488 1178 1360 1228">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="488 1239 1330 1289">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="488 1299 1260 1323">[Example: Consider a button which does not display an icon, as follows:</p> <div data-bbox="488 1333 727 1501" data-label="Image"> <p>The image shows a user interface with a button that has the text 'Button with no icon' and no icon. The button is part of a 'Custom Group'.</p> </div> <p data-bbox="488 1512 1109 1535">This would be specified using the following XML fragment:</p> <pre data-bbox="516 1562 987 1612" style="background-color: #f0f0f0;"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="488 1644 634 1667">end example]</p> <p data-bbox="488 1677 1317 1701">The possible values for this attribute are defined by the XML Schema boolean</p> |

| Attributes | Description |
|------------------------|---|
| | datatype. |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_VisibleButton">
  <complexContent>
    <restriction base="CT_ButtonRegular">
      <attribute name="visible" use="prohibited"/>
      <attribute name="isVisible" use="prohibited"/>
    </restriction>
  </complexContent>
</complexType>
```

2.2.5 buttonGroup (Button Grouping Container)

This element specifies a grouping container that groups controls together visually. The child controls are laid out horizontally.

[Example: Consider a group of buttons, as follows:



Figure 3: A group of buttons

This would be specified using the following XML fragment:

```
<buttonGroup id="buttonGroup">
  <button id="button1" imageMso="Bold" />
  <button id="button2" imageMso="Italic" />
  <button id="button3" imageMso="Underline" />
</buttonGroup>
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|-------------------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Unsize Menu) | §2.2.26 |
| splitButton (Unsize Split Button) | §2.2.36 |
| toggleButton (Unsize Toggle Button) | §2.2.42 |

| Attributes | Description |
|----------------------------------|--|
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these</p> |

| Attributes | Description |
|---|--|
| | <p>attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 289 1305 632"> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1073 1281 1150"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1514 1294 1591"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type</p> |

| Attributes | Description |
|---|--|
| | (§2.3.9). |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
complexType name="CT_ButtonGroup">
  <sequence>
    <choice minOccurs="0" maxOccurs="1000">
      <element name="control" type="CT_ControlCloneRegular"/>
      <element name="button" type="CT_ButtonRegular"/>
      <element name="toggleButton" type="CT_ToggleButtonRegular"/>
    </choice>
  </sequence>
</complexType>
```



```

<element name="gallery" type="CT_GalleryRegular"/>
<element name="menu" type="CT_MenuRegular"/>
<element name="dynamicMenu" type="CT_DynamicMenuRegular"/>
<element name="splitButton" type="CT_SplitButtonRegular"/>
</choice>
</sequence>
<attributeGroup ref="AG_IDCustom"/>
<attributeGroup ref="AG_Visible"/>
<attributeGroup ref="AG_PositionAttributes"/>
</complexType>

```

2.2.6 checkBox (Check Box)

This element specifies a standard checkbox control.

[Example: Consider a checkbox control, as follows:

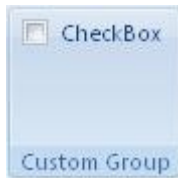


Figure 4: A checkbox control

This would be specified using the following XML fragment:

```
<checkBox id="checkBox" label="CheckBox" />
```

end example]

| Parent Elements |
|---|
| box (§2.2.1); group (§2.2.23); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); menu (§2.2.27); officeMenu (§2.2.31) |

| Attributes | Description |
|---------------------------|--|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="482 1505 1047 1608" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes</pre> |


| Attributes | Description |
|--|---|
| | <p>the function of this control in detail." /></p> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |


| Attributes | Description |
|--|--|
| | <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getPressed</code> (<code>getPressed</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the toggled state of this control.</p> <p>If this attribute is omitted the control should default to the off state.</p> <p>[Example: Consider the following XML fragment:</p> <pre><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p>This specifies that the <code>IsButtonToggled</code> callback function will be called when the application needs to determine the toggle state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> |



| Attributes | Description |
|--|---|
| | <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> (<code>getSupertip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> |

| Attributes | Description |
|-----------------------------|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1166 275" data-cs="2" data-kind="parent"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 642 1013 663" data-cs="2" data-kind="parent"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 984 821 1005" data-cs="2" data-kind="parent"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1331 1305 1671" data-cs="2" data-kind="parent"> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace</p> |

| Attributes | Description |
|--|--|
| | <p><code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>image</code> (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> |

| Attributes | Description |
|---|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> |

| Attributes | Description |
|------------------------------|---|
| | <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that</p> |

| Attributes | Description |
|---|---|
| | <p>the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>showLabel</code> (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> |

| Attributes | Description |
|------------------------------|---|
| | <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_CheckBox">
  <complexContent>
    <restriction base="CT_ToggleButtonRegular">
      <attribute name="image" use="prohibited"/>
      <attribute name="imageMso" use="prohibited"/>
      <attribute name="getImage" use="prohibited"/>
      <attribute name="showImage" use="prohibited"/>
      <attribute name="getShowImage" use="prohibited"/>
      <attribute name="showLabel" use="prohibited"/>
      <attribute name="getShowLabel" use="prohibited"/>
    </restriction>
  </complexContent>
</complexType>
```

2.2.7 comboBox (Combo Box)

This element specifies a standard combo box control that allows a user to input a text string or select one from a list.

[Example: Consider a combo box control, as follows:

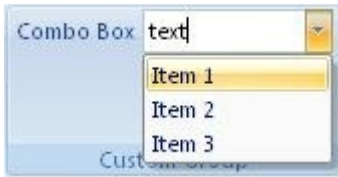


Figure 5: A combo box control

This would be specified using the following XML fragment:

```
<comboBox id="comboBox" label="Combo Box">
  <item id="item1" label="Item 1" />
  <item id="item2" label="Item 2" />
  <item id="item3" label="Item 3" />
</comboBox>
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|-----------------------|-----------|
| item (Selection Item) | §2.2.24 |

| Attributes | Description |
|-------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

| Attributes | Description |
|--|---|
| <code>getEnabled</code> (<code>getEnabled Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (<code>getImage Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemCount</code> (<code>getItemCount Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the number of selection items in this control.</p> <p>If this attribute is omitted the control should display any selection items which are specified as child elements. If no such items are specified, the control should be empty.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>This specifies that the <code>GetGalleryItemCount</code> callback function will be called when the application needs to determine the number of items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemID</code> (<code>getItemID Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the ID of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should have empty IDs.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>This specifies that the <code>GetGalleryItemID</code> callback function will be called when the application needs to determine the ID of a selection item. end example]</p> |

| Attributes | Description |
|---|---|
| | The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2). |
| getItemImage (getItemImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display icons.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>This specifies that the GetGalleryItemImage callback function will be called when the application needs to determine the icon of a selection item. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemLabel (getItemLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display labels.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>This specifies that the GetGalleryItemLabel callback function will be called when the application needs to determine the label of a selection item. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemScreentip (getItemScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should use their labels as their screentips, or display no screentips at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>This specifies that the GetGalleryItemScreentip callback function will be called when the application needs to determine the screentip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemSupertip (getItemSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not</p> |


| Attributes | Description |
|---|--|
| | <p>display supertips.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 285 1317 338"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>This specifies that the <code>GetGalleryItemSupertip</code> callback function will be called when the application needs to determine the supertip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 726 1235 747"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1115 1211 1136"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1503 1317 1524"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |


| Attributes | Description |
|--|--|
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> (<code>getSupertip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getText</code> (<code>getText</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the text that should be displayed in the control.</p> <p>[Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" getText="GetEditBoxText" /></pre> <p>This specifies that the <code>GetEditBoxText</code> callback function will be called when the application needs to determine the text to display in the control. end</p> |


| Attributes | Description |
|----------------------------------|---|
| | <p>example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"></pre> |


| Attributes | Description |
|--|--|
| | <pre> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|---|
| | <pre data-bbox="602 218 1365 296"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="573 325 1386 380">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p data-bbox="573 388 1373 443">The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p data-bbox="573 464 1386 518">Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="573 527 1395 627">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="573 636 1094 663">[Example: Consider the following XML fragment:</p> <pre data-bbox="602 688 1378 766"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="573 795 1386 850">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p data-bbox="573 858 1390 913">The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p data-bbox="573 932 1386 987">Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="573 995 1395 1096">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="573 1104 1094 1131">[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1157 1378 1234"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="573 1264 1386 1318">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p data-bbox="573 1327 1373 1381">The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p data-bbox="573 1398 1386 1453">Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="573 1461 1395 1562">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="573 1570 1094 1598">[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1623 1393 1701"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> |

| Attributes | Description |
|---|--|
| | <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>invalidateContentOnDrop</code> (Invalidate Content on Drop) | <p>Specifies whether this control should invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p>If this attribute is omitted its value should default to false.</p> <p>[Example: Consider the following XML fragment:</p> <pre><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>This specifies that this combo box should clear out its items and re-call the <code>GetComboBoxItemCount</code> and <code>GetComboBoxItemLabel</code> callback functions in order to populate its contents each time the user opens it. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>maxLength</code> (Maximum Input String Length) | <p>Specifies an integer which should be used as the maximum length of a string which can be entered into the control.</p> <p>If the <code>maxLength</code> attribute is omitted, the length of the input string should not be limited except by application-specific constraints.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|--|
| | <pre data-bbox="602 218 1097 241" style="background-color: #f0f0f0;"><editBox id="editBox" maxLength="10" /></pre> <p data-bbox="573 273 1344 325">This specifies an edit box control which can only accept strings up to 10 characters in length. end example]</p> <p data-bbox="573 331 1354 384">The possible values for this attribute are defined by the ST_StringLength simple type (§2.3.12).</p> |
| <p data-bbox="237 411 464 464">onChange (onChange Callback)</p> | <p data-bbox="573 411 1365 464">Specifies the name of a callback function which should be called when the text in the control has been changed by the user.</p> <p data-bbox="573 470 1094 493">[Example: Consider the following XML fragment:</p> <pre data-bbox="602 520 1292 543" style="background-color: #f0f0f0;"><editBox id="editBox" onChange="EditBoxTextChanged" /></pre> <p data-bbox="573 575 1386 627">This specifies an edit box control which would call the <code>EditBoxTextChanged</code> callback function when the user inputs a text string. end example]</p> <p data-bbox="573 634 1386 686">The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p data-bbox="237 711 480 735">screentip (Screentip)</p> | <p data-bbox="573 711 1365 821">Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="573 827 1187 850">[Example: Consider a button with a screentip, as follows:</p>  <p data-bbox="573 1205 1195 1228">This would be specified using the following XML fragment:</p> <pre data-bbox="602 1255 1300 1308" style="background-color: #f0f0f0;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="573 1339 721 1362">end example]</p> <p data-bbox="573 1369 1357 1421">The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p data-bbox="237 1451 513 1474">showImage (Show Image)</p> | <p data-bbox="573 1451 1138 1474">Specifies whether this control should display an icon.</p> <p data-bbox="573 1480 1393 1533">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="573 1539 1333 1591">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="573 1598 1344 1621">[Example: Consider a button which does not display an icon, as follows:</p> |

| Attributes | Description |
|--|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showItemImage (Show Item Image)</p> | <p>Specifies whether this control should display icons on its selection items. If this attribute is omitted the items' icons should be shown by default. [Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemImage="false" > <item id="item1" label="Item 1" /> <item id="item2" label="Item 1" /> <item id="item3" label="Item 2" /> <item id="item4" label="Item 3" /> </gallery></pre> <p>This specifies a gallery control which does not show any icons on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showLabel (Show Label)</p> | <p>Specifies whether this control should display its label. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>sizeString (Size String)</p> | <p>Specifies a string whose size should be used to determine the width of the text input area of this control. If this attribute is omitted the application should determine the width of the text input area of the control automatically. [Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" sizeString="WWWWWWWWWWWW" /></pre> <p>This specifies an edit box control which should be wide enough to display</p> |

| Attributes | Description |
|--|---|
| | <p>the string <code>XXXXXXXXXXXXXX</code>, end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>sizeString</code> (Size String)</p> | <p>Specifies a string whose size should be used to determine the width of the text input area of this control.</p> <p>If this attribute is omitted the application should determine the width of the text input area of the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" sizeString="XXXXXXXXXXXXXX" /></pre> <p>This specifies an edit box control which should be wide enough to display the string <code>XXXXXXXXXXXXXX</code>, end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>tag</code> (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> |

| Attributes | Description |
|---|---|
| | <p>This specifies a button with a tag value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ComboBox">
  <complexContent>
    <extension base="CT_EditBox">
      <sequence>
        <element name="item" type="CT_Item" minOccurs="0" maxOccurs="1000"/>
      </sequence>
      <attributeGroup ref="AG_DropDownAttributes"/>
      <attributeGroup ref="AG_DynamicContentAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.8 command (Repurposed Command)

This element specifies that a particular built-in command in the application should be repurposed.

The `enabled` and `getEnabled` attributes can be specified in order to disable a command.

The `onAction` attribute allows the functionality of a command to be repurposed to run a callback function. Only commands which execute simple actions (for example, commands represented as button controls) can be repurposed using `onAction`.

[Example: Consider the following XML fragment:

```
<commands>
  <command idMso="Bold" enabled="false" />
  <command idMso="Paste" onAction="MyPasteFunction" />
</commands>
```

This specifies that the `Bold` command should be permanently disabled and that the callback function `MyPasteFunction` should be called when the `Paste` command is invoked. end example]

| Parent Elements |
|-------------------------------------|
| commands (§2.2.9) |

| Attributes | Description |
|---|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> |

| Attributes | Description |
|------------|---|
| | The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2). |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Command" mixed="false">
  <attributeGroup ref="AG_Action"/>
  <attributeGroup ref="AG_Enabled"/>
  <attributeGroup ref="AG_IDMso"/>
</complexType>
```

2.2.9 commands (List of Repurposed Commands)

This element specifies a list of repurposed commands. This element should not be specified if the containing Custom UI XML document is a Quick Access Toolbar Customizations part.

| Parent Elements |
|--------------------|
| customUI (§2.2.14) |

| Child Elements | Subclause |
|------------------------------|-----------|
| command (Repurposed Command) | §2.2.8 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Commands">
  <sequence>
    <element name="command" type="CT_Command" minOccurs="1" maxOccurs="5000"/>
  </sequence>
</complexType>
```

2.2.10 contextualTabs (List of Contextual Tab Sets)

This element specifies a list of contextual tab sets. This element should not be specified if the containing Custom UI XML document is a Quick Access Toolbar Customizations part.

| Parent Elements |
|------------------|
| ribbon (§2.2.33) |

| Child Elements | Subclause |
|-----------------------------|-----------|
| tabSet (Contextual Tab Set) | §2.2.41 |

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_ContextualTabs">
  <sequence>
    <element name="tabSet" type="CT_TabSet" minOccurs="1" maxOccurs="100"/>
  </sequence>
</complexType>

```

2.2.11 control (Unsize Control Clone)

This element specifies a clone of a control which, due to its location, cannot have its size changed. The `size` attribute is not present. The element otherwise behaves like the regular `control` (§2.2.12) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); menu (§2.2.27); officeMenu (§2.2.31) |


| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|--|---|
| | <pre data-bbox="516 218 1128 243"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="488 273 1349 325">This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p data-bbox="488 331 1349 384">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p data-bbox="488 409 1349 462">Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p data-bbox="488 468 1349 520">The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p data-bbox="488 527 1003 552">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 581 1154 606"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="488 636 1349 688">This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p data-bbox="488 695 1349 747">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p data-bbox="488 766 1349 819">Specifies the name of a callback function which should be called to determine the label of this control.</p> <p data-bbox="488 825 1349 877">The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p data-bbox="488 884 1003 909">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 938 1128 963"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="488 993 1349 1045">This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p data-bbox="488 1052 1349 1104">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p data-bbox="488 1123 1349 1176">Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p data-bbox="488 1182 1349 1266">The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="488 1272 1003 1297">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1327 1230 1352"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="488 1381 1349 1434">This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p data-bbox="488 1440 1349 1493">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p data-bbox="488 1514 1349 1566">Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p data-bbox="488 1572 1349 1625">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="488 1631 1349 1684">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="488 1690 1003 1715">[Example: Consider the following XML fragment:</p> |


| Attributes | Description |
|---|--|
| | <pre data-bbox="516 218 1256 243"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="488 275 1382 352">This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p data-bbox="488 363 1357 415">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 436 399 510"><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p data-bbox="488 436 1321 489">Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p data-bbox="488 499 1360 552">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="488 562 1330 615">The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p data-bbox="488 625 1008 646">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 667 1256 693"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="488 724 1382 802">This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p data-bbox="488 812 1357 865">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 882 383 955"><code>getSupertip</code> (<code>getSupertip</code> Callback)</p> | <p data-bbox="488 882 1360 934">Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p data-bbox="488 945 1317 997">The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p data-bbox="488 1008 1008 1029">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1050 1208 1075"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="488 1106 1382 1159">This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p data-bbox="488 1169 1357 1222">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1243 367 1316"><code>getVisible</code> (<code>getVisible</code> Callback)</p> | <p data-bbox="488 1243 1360 1295">Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p data-bbox="488 1306 1382 1358">The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p data-bbox="488 1369 1008 1390">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1411 1167 1436"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="488 1467 1360 1520">This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p data-bbox="488 1530 1357 1583">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1604 399 1635"><code>id</code> (Control ID)</p> | <p data-bbox="488 1604 1382 1682">Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p data-bbox="488 1692 1325 1713">The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these</p> |

| Attributes | Description |
|---|--|
| | <p>attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 285 1015 310"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <p><code>idMso</code> (Built-in Control ID)</p> | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 632 824 657"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <p><code>idQ</code> (Qualified Control ID)</p> | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 978 1307 1318"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <p><code>image</code> (Custom Image ID)</p> | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|---|
| | <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> | <p>Specifies the ID of a built-in control which this control should be inserted before. If</p> |

| Attributes | Description |
|--|--|
| (ID of Built-in Control to Insert Before) | <p>the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> |

| Attributes | Description |
|--|---|
| | <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>showImage</code> (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>showLabel</code> (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that</p> |

| Attributes | Description |
|---|--|
| | <p>the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>tag</code> (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |

| Attributes | Description |
|---|---|
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ControlCloneRegular">
  <complexContent>
    <restriction base="CT_Control">
      <attribute name="id" use="prohibited"/>
    </restriction>
  </complexContent>
</complexType>
```

2.2.12 control (Control Clone)

This element specifies a clone of an existing control. Built-in controls can be cloned using the `idMso` attribute. Custom controls can be cloned using the `idQ` attribute. Custom controls cannot be created using the `control` element.

When an existing control is cloned, its non-location-specific properties such as the icon and label will be copied to the clone. Location-specific properties such as the size and visibility of the control will not be copied. These properties can be set by specifying additional attributes on the `control` element.

[Example: Consider the following XML fragment:

```
<control idMso="Paste" size="large" />
```

This will result in a large copy of the Paste control, as follows:

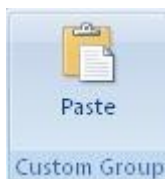



Figure 6: A Paste control

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Attributes | Description |
|---|--|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |


| Attributes | Description |
|--|--|
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> |



| Attributes | Description |
|---|---|
| Callback) | <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSize</code> (<code>getSize</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type</p> |



| Attributes | Description |
|--|--|
| | (§2.3.2). |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other</p> |

| Attributes | Description |
|--|---|
| | <p>Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>image</code> (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> |

| Attributes | Description |
|--|--|
| Control to Insert After) | <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 365 1284 443"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 806 1295 884"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertBeforeMso (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1268 1295 1346"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|---|
| | <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>onAction</code> (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>screentip</code> (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p> |

| Attributes | Description |
|-------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <ul style="list-style-type: none"> This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> |

| Attributes | Description |
|----------------------------|---|
| | <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>size (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>tag (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> |

| Attributes | Description |
|---|--|
| | <p>If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible. [Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:


```
<complexType name="CT_ControlClone">
  <complexContent>
    <restriction base="CT_Button">
      <attribute name="id" use="prohibited"/>
      <attribute name="onAction" use="prohibited"/>
    </restriction>
  </complexContent>
</complexType>
```

2.2.13 control (Quick Access Toolbar Control Clone)

This element specifies a clone of an existing control. It is specific to control clones on the quick access toolbar but otherwise behaves the same way as the regular `control` (§2.2.12) element.

| Parent Elements |
|--|
| <code>documentControls</code> (§2.2.16); <code>sharedControls</code> (§2.2.35) |

| Attributes | Description |
|--|---|
| <code>description</code> (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text. [Example: Consider a button with a detailed description, as follows:</p> |

| Attributes | Description |
|--|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |



| Attributes | Description |
|--------------------------------------|---|
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowImage (getShowImage) | <p>Specifies the name of a callback function which should be called to determine</p> |

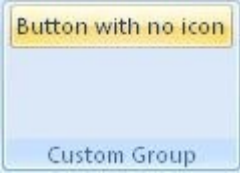

| Attributes | Description |
|--------------------------------------|---|
| Callback) | <p>whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSize (getSize Callback) | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> |


| Attributes | Description |
|-------------------------------------|---|
| | <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Custom Control ID) | <p>Specifies the ID for a custom control. All new custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control. The ID is qualified with an XML namespace prefix which specifies the owner of the control. If the namespace is equal to the Custom UI namespace, the <code>idQ</code> attribute behaves like the <code>idMso</code> attribute. If the namespace is equal to the name of the current file, the <code>idQ</code> attribute behaves like the <code>id</code> attribute. If the namespace is equal to the name of a different file, the attribute will reference a control from that file.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idQ="x:OtherTab"> <group id="MyGroup" label="My Group"> ... </group></pre> |

| Attributes | Description |
|--|--|
| | <p></tab></p> <p>In this case <code>x</code> is an XML namespace equal to the name of another file which has a Custom UI document with a <code>tab</code> with an ID of <code>OtherTab</code>. This example adds a custom <code>group</code> to that <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>image</code> (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom <code>tab</code> with an ID of <code>MyTab</code> should be inserted after the built-in <code>tab</code> with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> | <p>Specifies the qualified ID of a control which this control should be inserted after. If</p> |

| Attributes | Description |
|--|--|
| (Qualified ID of Control to Insert After) | <p>the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p> |

| Attributes | Description |
|-------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| <p>label (Label)</p> | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>screentip (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> |

| Attributes | Description |
|-------------------------------|---|
| | <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showLabel (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>size (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p> |

| Attributes | Description |
|------------------------------|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ControlCloneQat">
  <complexContent>
    <extension base="CT_ControlBase">
      <attribute name="id" type="ST_ID" use="optional"/>
      <attribute name="idQ" type="ST_QID" use="optional"/>
      <attributeGroup ref="AG_IDMso"/>
      <attributeGroup ref="AG_Description"/>
      <attributeGroup ref="AG_SizeAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.14 customUI (Custom UI Document Root)

This element specifies the root tag in a Custom UI XML document.

| Child Elements | Subclause |
|--|-------------------------|
| commands (List of Repurposed Commands) | §2.2.9 |
| ribbon (Ribbon) | §2.2.33 |

| Attributes | Description |
|--------------------------------------|--|
| loadImage (loadImage Callback) | <p>Specifies the name of a callback function that should be called when the application needs to load an image for a control's icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="..." loadImage="LoadImageFunction" /></pre> <p>This specifies that the LoadImageFunction callback should be called to load icon images. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| onLoad (onLoad Callback) | <p>Specifies the name of a callback function which should be called when the Custom UI file is loaded by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="..." onLoad="OnCustomUILoaded" /></pre> <p>This specifies that the OnCustomUILoaded callback function should be called when the containing Custom UI file is loaded. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_CustomUI">
  <sequence>
    <element name="commands" type="CT_Commands" minOccurs="0" maxOccurs="1"/>
    <element name="ribbon" type="CT_Ribbon" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="onLoad" type="ST_Delegate" use="optional"/>
  <attribute name="loadImage" type="ST_Delegate" use="optional"/>
</complexType>
```

2.2.15 dialogBoxLauncher (Dialog Box Launcher)

This element specifies a button which is the dialog box launcher control for a ribbon group.

[Example: Consider a dialog box launcher control, as follows:



Figure 7: A dialog box launcher control

This would be specified using the following XML fragment:

```
<group id="customGroup" label="Custom Group">
  <dialogBoxLauncher>
    <button id="button" screentip="Dialog Box Launcher" />
  </dialogBoxLauncher>
</group>
```

end example]

| Parent Elements |
|-----------------------------------|
| group (§2.2.23) |

| Child Elements | Subclause |
|------------------------|------------------------|
| button (Unsize Button) | §2.2.3 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_DialogLauncher">
  <sequence>
    <element name="button" type="CT_ButtonRegular" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

2.2.16 documentControls (List of Document-Specific Quick Access Toolbar Controls)

This element specifies the list of controls on the quick access toolbar which are specific to the containing file.

[Example: Consider a set of controls on the document-specific quick access toolbar, as follows:

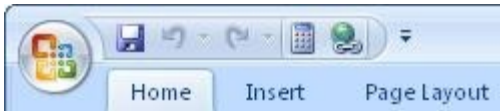


Figure 8: A set of controls on the document-specific quick access toolbar

This would be specified using the following XML fragment:

```
<documentControls>
  <control idMso="CalculateNow" />
  <control idMso="HyperlinkInsert" />
</documentControls>
```

end example]

| Parent Elements |
|---------------------------------|
| qat (§2.2.32) |

| Child Elements | Subclause |
|--|-------------------------|
| button (Unsize Button) | §2.2.3 |
| control (Quick Access Toolbar Control Clone) | §2.2.13 |
| separator (Separator) | §2.2.34 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_QatItems">
  <sequence>
    <choice minOccurs="0" maxOccurs="1000">
      <element name="control" type="CT_ControlCloneQat"/>
      <element name="button" type="CT_ButtonRegular"/>
      <element name="separator" type="CT_Separator"/>
    </choice>
  </sequence>
</complexType>
```

2.2.17 dropDown (Drop-down Control)

This element specifies a drop-down control that allows users to make a selection from a list of options. A drop-down control can optionally have buttons below its selection items.

[Example: Consider a drop-down control, as follows:

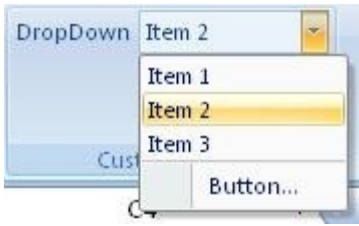


Figure 9: A drop-down control

This would be specified using the following XML fragment:

```
<dropDown id="dropDown" label="DropDown">
  <item id="item1" label="Item 1" />
  <item id="item2" label="Item 2" />
  <item id="item3" label="Item 3" />
  <button id="button" label="Button..." />
</dropDown>
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| item (Selection Item) | §2.2.24 |

| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> |

| Attributes | Description |
|---|---|
| | <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (<code>getImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemCount</code> (<code>getItemCount</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the number of selection items in this control.</p> <p>If this attribute is omitted the control should display any selection items which are specified as child elements. If no such items are specified, the control should be empty.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>This specifies that the <code>GetGalleryItemCount</code> callback function will be called when the application needs to determine the number of items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemID</code> (<code>getItemID</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the ID of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should have empty IDs.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>This specifies that the <code>GetGalleryItemID</code> callback function will be called when the application needs to determine the ID of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemImage</code> | Specifies the name of a callback function which should be called to determine |

| Attributes | Description |
|--|--|
| (getItemImage Callback) | <p>the icon of a specific dynamically-created selection item, identified by index. If this attribute is omitted, dynamically-created selection items should not display icons.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>This specifies that the <code>GetGalleryItemImage</code> callback function will be called when the application needs to determine the icon of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemLabel (getItemLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of a specific dynamically-created selection item, identified by index. If this attribute is omitted, dynamically-created selection items should not display labels.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>This specifies that the <code>GetGalleryItemLabel</code> callback function will be called when the application needs to determine the label of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemScreentip (getItemScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of a specific dynamically-created selection item, identified by index. If this attribute is omitted, dynamically-created selection items should use their labels as their screentips, or display no screentips at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>This specifies that the <code>GetGalleryItemScreentip</code> callback function will be called when the application needs to determine the screentip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemSupertip (getItemSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of a specific dynamically-created selection item, identified by index. If this attribute is omitted, dynamically-created selection items should not display supertips.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> |


| Attributes | Description |
|--|---|
| | <p>This specifies that the <code>GetGalleryItemSupertip</code> callback function will be called when the application needs to determine the supertip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSelectedItemID</code> (<code>getSelectedItemID Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the ID of the item which should be selected in this control.</p> <p>The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|--|---|
| | <pre data-bbox="565 218 1284 296"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p data-bbox="540 325 1393 428">This specifies that the <code>GetGallerySelectedItemID</code> callback function will be called when the application needs to determine the selected item in the gallery. In this example the callback function should return one of the IDs returned by the <code>GetItemID</code> callback function. end example]</p> <p data-bbox="540 436 1354 491">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSelectedItemIndex</code> (<code>getSelectedItemIndex</code> Callback) | <p data-bbox="540 514 1370 569">Specifies the name of a callback function which should be called to determine the index of the item which should be selected in this control.</p> <p data-bbox="540 575 1365 651">The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p data-bbox="540 659 1060 686">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 714 1284 762"><gallery id="gallery" getItemCount="GetGalleryItemCount" getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p data-bbox="540 791 1393 867">This specifies that the <code>GetGallerySelectedItemIndex</code> callback function will be called when the application needs to determine the selected item in the gallery. end example]</p> <p data-bbox="540 875 1354 930">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p data-bbox="540 955 1370 1010">Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p data-bbox="540 1016 1360 1064">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="540 1073 1382 1127">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="540 1136 1060 1163">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 1188 1308 1209"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="540 1239 1370 1314">This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p data-bbox="540 1323 1354 1377">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p data-bbox="540 1396 1370 1451">Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p data-bbox="540 1459 1360 1514">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="540 1522 1382 1577">The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p data-bbox="540 1583 1060 1610">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 1635 1308 1656"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="540 1686 1370 1711">This specifies that the <code>IsButtonLabelVisible</code> callback function will be called</p> |

| Attributes | Description |
|------------------------------------|---|
| | <p>when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type</p> |


| Attributes | Description |
|------------------------------|--|
| | (§2.3.5). |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case ex is an XML namespace prefix for the namespace http://www.example.com. This XML fragment refers to a tab in that namespace with an ID of OtherTab. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of ForestPic. end example]</p> <p>The possible values for this attribute are defined by the ST_Uri simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> |

| Attributes | Description |
|--|---|
| | <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |

| Attributes | Description |
|--|--|
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple</p> |

| Attributes | Description |
|---------------------------------|---|
| | type (§2.3.2). |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showItemImage (Show Item Image) | <p>Specifies whether this control should display icons on its selection items.</p> <p>If this attribute is omitted the items' icons should be shown by default.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|---|
| | <pre data-bbox="565 218 1349 373"><gallery id="gallery" label="Gallery" showItemImage="false" > <item id="item1" label="Item 1" /> <item id="item2" label="Item 1" /> <item id="item3" label="Item 2" /> <item id="item4" label="Item 3" /> </gallery></pre> <p data-bbox="540 405 1370 457">This specifies a gallery control which does not show any icons on its selection items. end example]</p> <p data-bbox="540 464 1370 516">The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showItemLabel (Show Item Label) | <p data-bbox="540 541 1325 594">Specifies whether this control should display labels on its selection items. If this attribute is omitted the item's labels should be shown by default.</p> <p data-bbox="540 611 1060 636">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 663 1349 819"><gallery id="gallery" label="Gallery" showItemLabel="false" > <item id="item1" image="Forest" /> <item id="item2" image="Desert" /> <item id="item3" image="Mountain" /> <item id="item4" image="Ocean" /> </gallery></pre> <p data-bbox="540 850 1325 903">This would specify a gallery control which does not show any labels on its selection items. end example]</p> <p data-bbox="540 909 1370 961">The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p data-bbox="540 982 1114 1008">Specifies whether this control should display its label.</p> <p data-bbox="540 1018 1360 1071">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="540 1077 1382 1129">The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p data-bbox="540 1136 1060 1161">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 1188 1219 1241"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="540 1272 1390 1325">This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p data-bbox="540 1331 1370 1383">The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| sizeString (Size String) | <p data-bbox="540 1402 1390 1455">Specifies a string whose size should be used to determine the width of the text input area of this control.</p> <p data-bbox="540 1461 1352 1514">If this attribute is omitted the application should determine the width of the text input area of the control automatically.</p> <p data-bbox="540 1520 1060 1545">[Example: Consider the following XML fragment:</p> <pre data-bbox="565 1572 1219 1598"><editBox id="editBox" sizeString="XXXXXXXXXXXXXXXX" /></pre> <p data-bbox="540 1629 1365 1682">This specifies an edit box control which should be wide enough to display the string <code>XXXXXXXXXXXXXXXX</code>. end example]</p> <p data-bbox="540 1688 1378 1713">The possible values for this attribute are defined by the <code>ST_String</code> simple type</p> |

| Attributes | Description |
|------------------------------|---|
| | (§2.3.11). |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control. The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean</p> |

| Attributes | Description |
|------------|-------------|
| | datatype. |

The following XML Schema fragment defines the contents of this element:


```
<complexType name="CT_DropDownRegular">
  <complexContent>
    <extension base="CT_Control">
      <sequence>
        <element name="item" type="CT_Item" minOccurs="0" maxOccurs="1000"/>
        <element name="button" type="CT_ButtonRegular" minOccurs="0" maxOccurs="16"/>
      </sequence>
      <attributeGroup ref="AG_Action"/>
      <attributeGroup ref="AG_Enabled"/>
      <attributeGroup ref="AG_Image"/>
      <attributeGroup ref="AG_DropDownAttributes"/>
      <attribute name="getSelectedItemID" type="ST_Delegate" use="optional"/>
      <attribute name="getSelectedItemIndex" type="ST_Delegate" use="optional"/>
      <attribute name="showItemLabel" type="xsd:boolean" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.18 dynamicMenu (UnsizeD Dynamic Menu)

This element specifies a dynamic menu control which, due to its location, cannot have its anchor size changed. The `size` attribute is not present. It otherwise behaves identically to the regular `dynamicMenu` (§2.2.19) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); menu (§2.2.27); officeMenu (§2.2.31) |

| Attributes | Description |
|---------------------------|---|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="568 1453 1144 1560" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> |

| Attributes | Description |
|--|--|
| | <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| <p>enabled (Enabled State)</p> | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>getContent</code> (GetContent Callback)</p> | <p>Specifies the name of a callback function which should be called when the application needs to determine the contents of the control.</p> <p>[Example: Consider a dynamic menu control, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" /></pre> <p>The <code>GetMenuContent</code> callback function is called when the menu is dropped, and in this case would return a string with the following XML:</p> <pre><menu xmlns="http://schemas.microsoft.com/office/2006/01/customui"> <button id="button1" label="Button 1" /> <button id="button2" label="Button 2" /> <button id="button3" label="Button 3" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p><code>getDescription</code> (getDescription Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> |


| Attributes | Description |
|--|--|
| | <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getEnabled</code> (<code>getEnabled Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (<code>getImage Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel Callback</code>) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> |


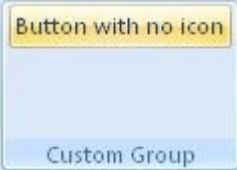
| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 254 1211 275" data-cs="2" data-kind="parent"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getScreentip</code> (<code>getScreentip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 642 1313 663" data-cs="2" data-kind="parent"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowImage</code> (<code>getShowImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1089 1338 1110" data-cs="2" data-kind="parent"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1539 1338 1560" data-cs="2" data-kind="parent"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |


| Attributes | Description |
|---|---|
| <code>getSupertip</code> (<code>getSupertip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown. [Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible. [Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified. [Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified. [Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |

| Attributes | Description |
|------------------------------|--|
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> |

| Attributes | Description |
|--|--|
| | The possible values for this attribute are defined by the ST_ID simple type (§2.3.5). |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | Specifies the qualified ID of a control which this control should be inserted before. |

| Attributes | Description |
|---|--|
| Before) | <p>before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| invalidateContentOnDrop (Invalidate Content on Drop) | <p>Specifies whether this control should invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p>If this attribute is omitted its value should default to false.</p> <p>[Example: Consider the following XML fragment:</p> <pre><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>This specifies that this combo box should clear out its items and re-call the <code>GetComboBoxItemCount</code> and <code>GetComboBoxItemLabel</code> callback functions in order to populate its contents each time the user opens it. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|-------------------------------------|--|
| | <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>screentip</code> (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>showImage</code> (Show Image) | <p>Specifies whether this control should display an icon. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

| Attributes | Description |
|-------------------------------|---|
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of 123456, which will be passed to</p> |

| Attributes | Description |
|---|---|
| | <p>the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_DynamicMenuRegular">
  <complexContent>
    <extension base="CT_ControlBase">
      <attributeGroup ref="AG_Description"/>
      <attributeGroup ref="AG_IDAttributes"/>
      <attributeGroup ref="AG_GetContentAttributes"/>
      <attributeGroup ref="AG_DynamicContentAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.19 dynamicMenu (Dynamic Menu)

This element specifies a dynamic menu control which populates its contents dynamically.

[Example: Consider a dynamic menu control, as follows:



Figure 10: A dynamic menu control

This would be specified using the following XML fragment:

```
<dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" />
```

The `GetMenuContent` callback function is called when the menu is dropped, and in this case would return a string with the following XML:

```


<menu xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <button id="button1" label="Button 1" />
  <button id="button2" label="Button 2" />
  <button id="button3" label="Button 3" />
</menu>

```

end example]

| Parent Elements |
|---|
| box (§2.2.1); group (§2.2.23) |

| Attributes | Description |
|----------------------------------|--|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> <div data-bbox="568 812 1136 917" data-label="Image"> </div> <p>This would be specified using the following XML fragment:</p> <pre> <button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /> </pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" label="Disabled Button" enabled="false" /> </pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getContent (GetContent Callback) | <p>Specifies the name of a callback function which should be called when the application needs to determine the contents of the control.</p> <p>[Example: Consider a dynamic menu control, as follows:</p> |

| Attributes | Description |
|--|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" /></pre> <p>The <code>GetMenuContent</code> callback function is called when the menu is dropped, and in this case would return a string with the following XML:</p> <pre><menu xmlns="http://schemas.microsoft.com/office/2006/01/customui"> <button id="button1" label="Button 1" /> <button id="button2" label="Button 2" /> <button id="button3" label="Button 3" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getDescription</code> (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getEnabled</code> (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |


| Attributes | Description |
|---|--|
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple</p> |


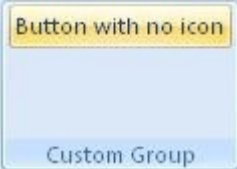
| Attributes | Description |
|---|--|
| | type (§2.3.2). |
| getShowImage (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSize (getSize Callback) | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> |



| Attributes | Description |
|---|--|
| | <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI</pre> |

| Attributes | Description |
|--|--|
| | <pre> xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order</p> |

| Attributes | Description |
|--|--|
| | <p>they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 285 1365 359"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 751 1365 825"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1220 1365 1293"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1686 1365 1707"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"></pre> |

| Attributes | Description |
|---|--|
| | <p>...</p> <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>invalidateContentOnDrop</code> (Invalidate Content on Drop) | <p>Specifies whether this control should invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p>If this attribute is omitted its value should default to false.</p> <p>[Example: Consider the following XML fragment:</p> <pre><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>This specifies that this combo box should clear out its items and re-call the <code>GetComboBoxItemCount</code> and <code>GetComboBoxItemLabel</code> callback functions in order to populate its contents each time the user opens it. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>screentip</code> (Screentip) | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the</p> |

| Attributes | Description |
|------------------------|---|
| | <p>control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false"</pre> |

| Attributes | Description |
|----------------------------|---|
| | <pre>imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>size (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>tag (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_DynamicMenu">
  <complexContent>
    <extension base="CT_DynamicMenuRegular">
      <attributeGroup ref="AG_SizeAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.20 editBox (Edit Box)

This element specifies an edit box control that allows a user to enter a string of text.

[Example: Consider an edit box control, as follows:

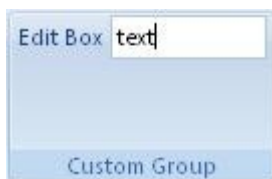


Figure 11: An edit box control

This would be specified using the following XML fragment:

```
<editBox id="editBox" label="Edit Box" />
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |


| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> |


| Attributes | Description |
|---|--|
| Callback) | <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> | Specifies the name of a callback function which should be called to determine |


| Attributes | Description |
|------------------------------------|---|
| (getShowLabel Callback) | <p>whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getText (getText Callback) | <p>Specifies the name of a callback function which should be called to determine the text that should be displayed in the control.</p> <p>[Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" getText="GetEditBoxText" /></pre> <p>This specifies that the <code>GetEditBoxText</code> callback function will be called when the application needs to determine the text to display in the control. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> |


| Attributes | Description |
|--|---|
| | <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>image</code> (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> |

| Attributes | Description |
|--|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1024 275" data-cs="2" data-kind="parent"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 646 1000 667" data-cs="2" data-kind="parent"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1037 1279 1108" data-cs="2" data-kind="parent"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1478 1292 1549" data-cs="2" data-kind="parent"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |

| Attributes | Description |
|---|--|
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|---|
| | <pre data-bbox="516 218 1078 239"><button id="button" label="Custom Button" /></pre> <p data-bbox="488 273 1305 294">This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p data-bbox="488 306 1325 357">The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>maxLength</code> (Maximum Input String Length) | <p data-bbox="488 384 1382 434">Specifies an integer which should be used as the maximum length of a string which can be entered into the control.</p> <p data-bbox="488 445 1357 495">If the <code>maxLength</code> attribute is omitted, the length of the input string should not be limited except by application-specific constraints.</p> <p data-bbox="488 506 1008 527">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 556 1013 577"><editBox id="editBox" maxLength="10" /></pre> <p data-bbox="488 611 1377 661">This specifies an edit box control which can only accept strings up to 10 characters in length. end example]</p> <p data-bbox="488 672 1344 722">The possible values for this attribute are defined by the <code>ST_StringLength</code> simple type (§2.3.12).</p> |
| <code>onChange</code> (onChange Callback) | <p data-bbox="488 747 1357 798">Specifies the name of a callback function which should be called when the text in the control has been changed by the user.</p> <p data-bbox="488 808 1008 829">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 858 1203 879"><editBox id="editBox" onChange="EditBoxTextChanged" /></pre> <p data-bbox="488 913 1393 963">This specifies an edit box control which would call the <code>EditBoxTextChanged</code> callback function when the user inputs a text string. end example]</p> <p data-bbox="488 974 1357 1024">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>screentip</code> (Screentip) | <p data-bbox="488 1050 1273 1071">Specifies a string which should be shown as the screentip for this control.</p> <p data-bbox="488 1081 1352 1161">The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="488 1171 1101 1192">[Example: Consider a button with a screentip, as follows:</p>  <p data-bbox="488 1541 1109 1562">This would be specified using the following XML fragment:</p> <pre data-bbox="516 1591 1214 1642"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="488 1675 634 1696">end example]</p> <p data-bbox="488 1707 1325 1728">The possible values for this attribute are defined by the <code>ST_String</code> simple type</p> |

| Attributes | Description |
|--------------------------|---|
| | (§2.3.11). |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| sizeString (Size String) | <p>Specifies a string whose size should be used to determine the width of the text input area of this control.</p> <p>If this attribute is omitted the application should determine the width of the text input area of the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" sizeString="WWWWWWWWWWWWW" /></pre> <p>This specifies an edit box control which should be wide enough to display the string WWWWWWWWWWWWWW. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>attribute is specified no supertip for this control should be shown. [Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control. If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control. The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible. [Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_EditBox">
  <complexContent>
    <extension base="CT_Control">
      <attributeGroup ref="AG_Enabled"/>
      <attributeGroup ref="AG_Image"/>
      <attribute name="maxLength" type="ST_StringLength" use="optional"/>
      <attribute name="getText" type="ST_Delegate" use="optional"/>
      <attribute name="onChange" type="ST_Delegate" use="optional"/>
      <attribute name="sizeString" type="ST_String" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

2.2.21 gallery (Gallery)

This element specifies a gallery control, which displays a drop-down grid of items that the user can select from. A gallery can optionally have buttons below its selection items.

[Example: Consider a gallery control that shows a selection of pictures, as follows:

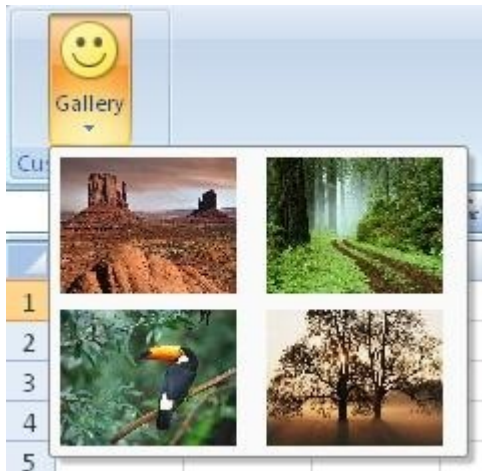


Figure 12: A gallery control

This would be specified using the following XML fragment:

```

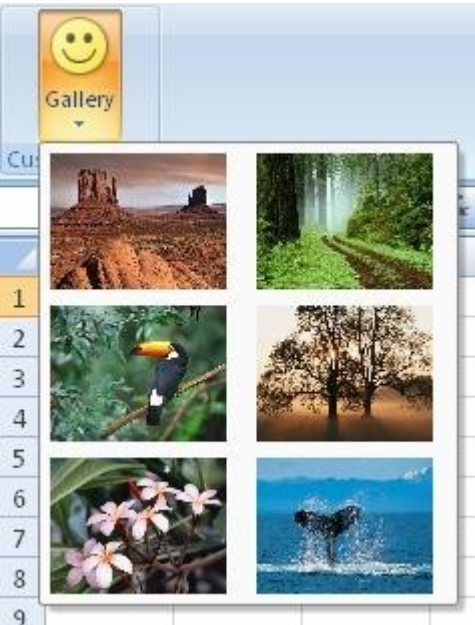
<gallery id="gallery" label="Gallery" itemWidth="88" itemHeight="68"
  size="large" imageMso="HappyFace" >
  <item id="item1" image="Desert" />
  <item id="item2" image="Forest" />
  <item id="item3" image="Toucan" />
  <item id="item4" image="Tree" />
</gallery>


```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|-------------------------------------|-------------------------|
| <code>button</code> (Unsize Button) | §2.2.3 |
| <code>item</code> (Selection Item) | §2.2.24 |

| Attributes | Description |
|-------------------------------------|--|
| <code>columns</code> (Column Count) | <p>Specifies the number of columns that the gallery's items should be arranged into.</p> <p>If the <code>columns</code> attribute is omitted the application should choose the number of columns automatically based on the number of items.</p> <p>[Example: Consider a gallery control with six items arranged into two columns, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" columns="2" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> <item id="item5" image="Flowers" /> <item id="item6" image="Whale" /> </gallery></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_GalleryRowColumnCount</code> simple type (§2.3.4).</p> |

| Attributes | Description |
|--|--|
| <code>description</code> (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| <code>enabled</code> (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <code>getDescription</code> (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <code>getEnabled</code> (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> |

| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 254 1252 275" data-cs="2" data-kind="parent"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getImage</code> (<code>getImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 642 1211 663" data-cs="2" data-kind="parent"><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getItemCount</code> (<code>getItemCount</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the number of selection items in this control.</p> <p>If this attribute is omitted the control should display any selection items which are specified as child elements. If no such items are specified, the control should be empty.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1031 1349 1052" data-cs="2" data-kind="parent"><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>This specifies that the <code>GetGalleryItemCount</code> callback function will be called when the application needs to determine the number of items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getItemHeight</code> (<code>getItemHeight</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the height of the selection items in this control.</p> <p>The <code>itemHeight</code> and <code>getItemHeight</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>getItemHeight</code> and <code>getItemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1503 1317 1577" data-cs="2" data-kind="parent"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>This specifies that the <code>GetGalleryItemHeight</code> callback function should be called when the application needs to determine the height of the items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple</p> |

| Attributes | Description |
|--|--|
| | type (§2.3.2). |
| getItemID (getItemID Callback) | <p>Specifies the name of a callback function which should be called to determine the ID of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should have empty IDs.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>This specifies that the <code>GetGalleryItemID</code> callback function will be called when the application needs to determine the ID of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemImage (getItemImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display icons.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>This specifies that the <code>GetGalleryItemImage</code> callback function will be called when the application needs to determine the icon of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemLabel (getItemLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display labels.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>This specifies that the <code>GetGalleryItemLabel</code> callback function will be called when the application needs to determine the label of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getItemScreentip (getItemScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should use their labels as their screentips, or display no screentips at all.</p> |

| Attributes | Description |
|--|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 254 1317 306"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>This specifies that the <code>GetGalleryItemScreentip</code> callback function will be called when the application needs to determine the screentip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemSupertip</code> (<code>getItemSupertip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display supertips.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 695 1317 747"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>This specifies that the <code>GetGalleryItemSupertip</code> callback function will be called when the application needs to determine the supertip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getItemWidth</code> (<code>getItemWidth</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the width of the selection items in this control.</p> <p>The <code>itemWidth</code> and <code>getItemWidth</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>getItemHeight</code> and <code>getItemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1192 1317 1272"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>This specifies that the <code>GetGalleryItemWidth</code> callback function should be called when the application needs to determine the width of the items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1661 1235 1682"><button id="button" getKeytip="GetButtonKeytip" /></pre> |

| Attributes | Description |
|---|--|
| | <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the <code>KeyTip</code> of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSelectedItemID</code> (<code>getSelectedItemID</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the ID of the item which should be selected in this control.</p> <p>The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p>This specifies that the <code>GetGallerySelectedItemID</code> callback function will be called when the application needs to determine the selected item in the gallery. In this example the callback function should return one of the IDs returned by the <code>GetItemID</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSelectedItemIndex</code> (<code>getSelectedItemIndex</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the index of the item which should be selected in this control.</p> <p>The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are</p> |


| Attributes | Description |
|---|--|
| | <p>mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 310 1317 363"><gallery id="gallery" getItemCount="GetGalleryItemCount" getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p>This specifies that the <code>GetGallerySelectedItemIndex</code> callback function will be called when the application needs to determine the selected item in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowImage</code> (<code>getShowImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 783 1341 810"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1230 1341 1257"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSize</code> (<code>getSize</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1619 1187 1646"><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> |

| Attributes | Description |
|------------------------------------|---|
| | The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2). |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type</p> |



| Attributes | Description |
|------------------------------|---|
| | (§2.3.5). |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case ex is an XML namespace prefix for the namespace http://www.example.com. This XML fragment refers to a tab in that namespace with an ID of OtherTab. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of ForestPic. end example]</p> <p>The possible values for this attribute are defined by the ST_Uri simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> |

| Attributes | Description |
|--|---|
| | <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |



| Attributes | Description |
|---|---|
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| invalidateContentOnDrop (Invalidate Content on Drop) | <p>Specifies whether this control should invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p>If this attribute is omitted its value should default to false.</p> <p>[Example: Consider the following XML fragment:</p> <pre><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>This specifies that this combo box should clear out its items and re-call the <code>GetComboBoxItemCount</code> and <code>GetComboBoxItemLabel</code> callback functions in order to populate its contents each time the user opens it. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| itemHeight (Selection Item Height) | <p>Specifies the height of the selection items in this control.</p> <p>The <code>itemHeight</code> and <code>getItemHeight</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>itemHeight</code> and <code>itemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider a gallery control with 68 pixel tall items. This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" itemWidth="88" itemHeight="68" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> </gallery></pre> <p>▪ end example]</p> <p>The possible values for this attribute are defined by the <code>ST_GalleryItemWidthHeight</code> simple type (§2.3.3).</p> |

| Attributes | Description |
|---|--|
| <code>itemWidth</code> (Selection Item Width) | <p>Specifies the width of the selection items in this control.</p> <p>The <code>itemWidth</code> and <code>getItemWidth</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>itemHeight</code> and <code>itemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider a gallery control with 88 pixel wide items. This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" itemWidth="88" itemHeight="68" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> </gallery></pre> <p>▪ end example]</p> <p>The possible values for this attribute are defined by the <code>ST_GalleryItemWidthHeight</code> simple type (§2.3.3).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>onAction</code> (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|--|
| | <pre data-bbox="602 218 1393 241" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p data-bbox="573 273 1393 323">This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p data-bbox="573 331 1393 382">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 411 431 434"><code>rows</code> (Row Count)</p> | <p data-bbox="573 411 1393 459">Specifies the number of rows that the gallery's items should be arranged into.</p> <p data-bbox="573 468 1393 518">If the <code>rows</code> attribute is omitted the application should choose the number of rows automatically based on the number of items.</p> <p data-bbox="573 527 1393 577">[Example: Consider a gallery control with six items arranged into two rows, as follows:</p> <div data-bbox="573 585 1248 1047" data-label="Image"> </div> <p data-bbox="573 1056 1195 1081">This would be specified using the following XML fragment:</p> <pre data-bbox="602 1108 1187 1346" style="background-color: #f0f0f0; padding: 5px;"> <gallery id="gallery" label="Gallery" rows="2" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> <item id="item5" image="Flowers" /> <item id="item6" image="Whale" /> </gallery> </pre> <p data-bbox="573 1375 721 1400">end example]</p> <p data-bbox="573 1409 1169 1459">The possible values for this attribute are defined by the <code>ST_GalleryRowColumnCount</code> simple type (§2.3.4).</p> |
| <p data-bbox="237 1488 480 1512"><code>screentip</code> (Screentip)</p> | <p data-bbox="573 1488 1393 1514">Specifies a string which should be shown as the screentip for this control.</p> <p data-bbox="573 1522 1393 1598">The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="573 1606 1187 1631">[Example: Consider a button with a screentip, as follows:</p> |

| Attributes | Description |
|--|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="602 596 1300 646"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="602 1205 1073 1255"><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showItemImage (Show Item Image)</p> | <p>Specifies whether this control should display icons on its selection items. If this attribute is omitted the items' icons should be shown by default.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1514 1382 1675"><gallery id="gallery" label="Gallery" showItemImage="false" > <item id="item1" label="Item 1" /> <item id="item2" label="Item 1" /> <item id="item3" label="Item 2" /> <item id="item4" label="Item 3" /> </gallery></pre> <p>This specifies a gallery control which does not show any icons on its</p> |

| Attributes | Description |
|---------------------------------|--|
| | <p>selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showItemLabel (Show Item Label) | <p>Specifies whether this control should display labels on its selection items. [Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemLabel="false" > <item id="item1" image="Forest" /> <item id="item2" image="Desert" /> <item id="item3" image="Mountain " /> <item id="item4" image="Ocean" /> </gallery></pre> <p>This would specify a gallery control which does not show any labels on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showItemLabel (Show Item Label) | <p>Specifies whether this control should display labels on its selection items. If this attribute is omitted the item's labels should be shown by default. [Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemLabel="false" > <item id="item1" image="Forest" /> <item id="item2" image="Desert" /> <item id="item3" image="Mountain " /> <item id="item4" image="Ocean" /> </gallery></pre> <p>This would specify a gallery control which does not show any labels on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| size (Control Size) | <p>Specifies the size of the control. The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size. [Example: Consider a large button, as follows:</p> |

| Attributes | Description |
|---------------------------------|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>sizeString (Size String)</p> | <p>Specifies a string whose size should be used to determine the width of the text input area of this control.</p> <p>If this attribute is omitted the application should determine the width of the text input area of the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" sizeString="WWWWWWWWWWWW" /></pre> <p>This specifies an edit box control which should be wide enough to display the string WWWWWWWWWWWWW. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |

| Attributes | Description |
|------------------------------|---|
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Gallery">
  <complexContent>
    <extension base="CT_GalleryRegular">
      <attributeGroup ref="AG_SizeAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

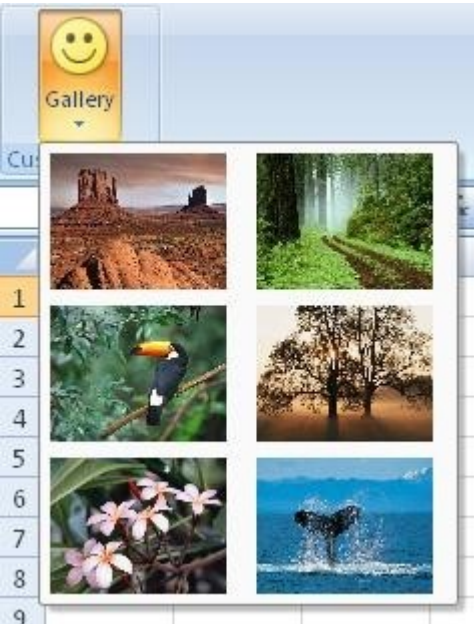
2.2.22 gallery (Unsize Gallery)


This element specifies a gallery which, due to its location, cannot have its size changed. The size attribute is not present. It otherwise behaves identically to the regular gallery (§2.2.21) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); menu (§2.2.27); officeMenu (§2.2.31) |

| Child Elements | Subclause |
|------------------------|-----------|
| button (Unsize Button) | §2.2.3 |

| Child Elements | Subclause |
|-----------------------|-------------------------|
| item (Selection Item) | §2.2.24 |

| Attributes | Description |
|---------------------------|--|
| columns (Column Count) | <p>Specifies the number of columns that the gallery's items should be arranged into.</p> <p>If the <code>columns</code> attribute is omitted the application should choose the number of columns automatically based on the number of items.</p> <p>[Example: Consider a gallery control with six items arranged into two columns, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" columns="2" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> <item id="item5" image="Flowers" /> <item id="item6" image="Whale" /> </gallery></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_GalleryRowColumnCount</code> simple type (§2.3.4).</p> |
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed</p> |

| Attributes | Description |
|--|---|
| | <p>text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called</p> |

| Attributes | Description |
|--|--|
| | <p>when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The getImage, image and imageMso attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the GetButtonImage callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemCount (getItemCount Callback) | <p>Specifies the name of a callback function which should be called to determine the number of selection items in this control.</p> <p>If this attribute is omitted the control should display any selection items which are specified as child elements. If no such items are specified, the control should be empty.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>This specifies that the GetGalleryItemCount callback function will be called when the application needs to determine the number of items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemHeight (getItemHeight Callback) | <p>Specifies the name of a callback function which should be called to determine the height of the selection items in this control.</p> <p>The itemHeight and getItemHeight attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The getItemHeight and getItemWidth attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>This specifies that the GetGalleryItemHeight callback function should be called when the application needs to determine the height of the items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getItemID (getItemID Callback) | <p>Specifies the name of a callback function which should be called to determine the ID of a specific dynamically-created selection item, identified by index.</p> |

| Attributes | Description |
|---|---|
| | <p>If this attribute is omitted, dynamically-created selection items should have empty IDs.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 310 1317 363"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>This specifies that the <code>GetGalleryItemID</code> callback function will be called when the application needs to determine the ID of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getItemImage</code> (<code>getItemImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the icon of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display icons.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 751 1317 804"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>This specifies that the <code>GetGalleryItemImage</code> callback function will be called when the application needs to determine the icon of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getItemLabel</code> (<code>getItemLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the label of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display labels.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1192 1317 1245"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>This specifies that the <code>GetGalleryItemLabel</code> callback function will be called when the application needs to determine the label of a selection item. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getItemScreentip</code> (<code>getItemScreentip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the screentip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should use their labels as their screentips, or display no screentips at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="602 1633 1317 1686"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>This specifies that the <code>GetGalleryItemScreentip</code> callback function will be</p> |

| Attributes | Description |
|---|---|
| | <p>called when the application needs to determine the screentip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getItemSupertip (getItemSupertip Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items should not display supertips.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 533 1317 583"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>This specifies that the <code>GetGalleryItemSupertip</code> callback function will be called when the application needs to determine the supertip of a selection item. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getItemWidth (getItemWidth Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the width of the selection items in this control.</p> <p>The <code>itemWidth</code> and <code>getItemWidth</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>getItemHeight</code> and <code>getItemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1035 1317 1106"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>This specifies that the <code>GetGalleryItemWidth</code> callback function should be called when the application needs to determine the width of the items in the gallery. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getKeytip (getKeytip Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="597 1499 1240 1524"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |

| Attributes | Description |
|--|--|
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSelectedItemID (getSelectedItemID Callback) | <p>Specifies the name of a callback function which should be called to determine the ID of the item which should be selected in this control.</p> <p>The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p>This specifies that the <code>GetGallerySelectedItemID</code> callback function will be called when the application needs to determine the selected item in the gallery. In this example the callback function should return one of the IDs returned by the <code>GetItemID</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSelectedItemIndex (getSelectedItemIndex Callback) | <p>Specifies the name of a callback function which should be called to determine the index of the item which should be selected in this control.</p> <p>The <code>getSelectedItemID</code> and <code>getSelectedItemIndex</code> attributes are mutually exclusive. If neither attribute is specified the control should not display a selected item.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" getItemCount="GetGalleryItemCount"</pre> |


| Attributes | Description |
|--|---|
| | <pre>getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p>This specifies that the <code>GetGallerySelectedItemIndex</code> callback function will be called when the application needs to determine the selected item in the gallery. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> (<code>getSupertip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> | <p>Specifies the name of a callback function which should be called to</p> |

| Attributes | Description |
|-----------------------------|---|
| Callback) | <p>determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ...</pre> |


| Attributes | Description |
|--|--|
| | <pre> </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> |

| Attributes | Description |
|--|--|
| | <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |


| Attributes | Description |
|--|---|
| invalidateContentOnDrop (Invalidate Content on Drop) | <p>Specifies whether this control should invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p>If this attribute is omitted its value should default to false.</p> <p>[Example: Consider the following XML fragment:</p> <pre><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>This specifies that this combo box should clear out its items and re-call the <code>GetComboBoxItemCount</code> and <code>GetComboBoxItemLabel</code> callback functions in order to populate its contents each time the user opens it. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| itemHeight (Selection Item Height) | <p>Specifies the height of the selection items in this control.</p> <p>The <code>itemHeight</code> and <code>getItemHeight</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>itemHeight</code> and <code>itemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider a gallery control with 68 pixel tall items. This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" itemWidth="88" itemHeight="68" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> </gallery></pre> <p>▪ end example]</p> <p>The possible values for this attribute are defined by the <code>ST_GalleryItemWidthHeight</code> simple type (§2.3.3).</p> |
| itemWidth (Selection Item Width) | <p>Specifies the width of the selection items in this control.</p> <p>The <code>itemWidth</code> and <code>getItemWidth</code> attributes are mutually exclusive. If neither attribute is specified the items should all take the size of the first item, based on its contents.</p> <p>The <code>itemHeight</code> and <code>itemWidth</code> attributes are mutually required. If only one of the attributes is specified, its value will be ignored.</p> <p>[Example: Consider a gallery control with 88 pixel wide items. This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" itemWidth="88" itemHeight="68" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> </gallery></pre> |

| Attributes | Description |
|------------------------------|--|
| | <ul style="list-style-type: none"> ▪ end example] <p>The possible values for this attribute are defined by the ST_GalleryItemWidthHeight simple type (§2.3.3).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| rows (Row Count) | <p>Specifies the number of rows that the gallery's items should be arranged into.</p> <p>If the <code>rows</code> attribute is omitted the application should choose the number of rows automatically based on the number of items.</p> <p>[Example: Consider a gallery control with six items arranged into two rows, as follows:</p> |

| Attributes | Description |
|------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" rows="2" size="large" imageMso="HappyFace" > <item id="item1" image="Desert" /> <item id="item2" image="Forest" /> <item id="item3" image="Toucan" /> <item id="item4" image="Tree" /> <item id="item5" image="Flowers" /> <item id="item6" image="Whale" /> </gallery></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_GalleryRowColumnCount simple type (§2.3.4).</p> |
| <p>screentip (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> |

| Attributes | Description |
|--|---|
| | <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showItemImage (Show Item Image)</p> | <p>Specifies whether this control should display icons on its selection items. If this attribute is omitted the items' icons should be shown by default.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemImage="false" > <item id="item1" label="Item 1" /> <item id="item2" label="Item 1" /> <item id="item3" label="Item 2" /> <item id="item4" label="Item 3" /> </gallery></pre> <p>This specifies a gallery control which does not show any icons on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showItemLabel (Show Item Label)</p> | <p>Specifies whether this control should display labels on its selection items.</p> <p>[Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemLabel="false" > <item id="item1" image="Forest" /> <item id="item2" image="Desert" /> <item id="item3" image="Mountain" /> <item id="item4" image="Ocean" /> </gallery></pre> <p>This would specify a gallery control which does not show any labels on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema</p> |

| Attributes | Description |
|---------------------------------|--|
| | boolean datatype. |
| showItemLabel (Show Item Label) | <p>Specifies whether this control should display labels on its selection items. If this attribute is omitted the item's labels should be shown by default. [Example: Consider the following XML fragment:</p> <pre><gallery id="gallery" label="Gallery" showItemLabel="false" > <item id="item1" image="Forest" /> <item id="item2" image="Desert" /> <item id="item3" image="Mountain" /> <item id="item4" image="Ocean" /> </gallery></pre> <p>This would specify a gallery control which does not show any labels on its selection items. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| sizeString (Size String) | <p>Specifies a string whose size should be used to determine the width of the text input area of this control. If this attribute is omitted the application should determine the width of the text input area of the control automatically. [Example: Consider the following XML fragment:</p> <pre><editBox id="editBox" sizeString="WWWWWWWWWWWWW" /></pre> <p>This specifies an edit box control which should be wide enough to display the string <code>WWWWWWWWWWWWW</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control. The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown. [Example: Consider a control with a supertip, as follows:</p> |

| Attributes | Description |
|------------------------------|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_GalleryRegular">
  <complexContent>
```

```

<extension base="CT_DropDownRegular">
<attributeGroup ref="AG_Description"/>
<attributeGroup ref="AG_DynamicContentAttributes"/>
<attribute name="columns" type="ST_GalleryRowColumnCount" use="optional"/>
<attribute name="rows" type="ST_GalleryRowColumnCount" use="optional"/>
<attribute name="itemWidth" type="ST_GalleryItemWidthHeight" use="optional"/>
<attribute name="itemHeight" type="ST_GalleryItemWidthHeight" use="optional"/>
<attribute name="getItemWidth" type="ST_Delegate" use="optional"/>
<attribute name="getItemHeight" type="ST_Delegate" use="optional"/>
<attribute name="showItemLabel" type="xsd:boolean" use="optional"/>
</extension>
</complexContent>
</complexType>

```

2.2.23 group (Group)

This element specifies a grouping of controls on a ribbon tab. All controls displayed in a ribbon tab must be contained within a `group`.

[Example: Consider a group with a single button, as follows:



Figure 13: A group with a single button

This would be specified using the following XML fragment:

```

<group id="group" label="Custom Group">
  <button id="button" label="Button" imageMso="HappyFace" />
</group>

```

end example]

| Parent Elements |
|---------------------------------|
| tab (§2.2.39) |

| Child Elements | Subclause |
|---|------------------------|
| box (Box Grouping Container) | §2.2.1 |
| button (Button) | §2.2.2 |
| buttonGroup (Button Grouping Container) | §2.2.5 |
| checkBox (Check Box) | §2.2.6 |
| comboBox (Combo Box) | §2.2.7 |


| Child Elements | Subclause |
|---|-------------------------|
| control (Control Clone) | §2.2.12 |
| dialogBoxLauncher (Dialog Box Launcher) | §2.2.15 |
| dropDown (Drop-down Control) | §2.2.17 |
| dynamicMenu (Dynamic Menu) | §2.2.19 |
| editBox (Edit Box) | §2.2.20 |
| gallery (Gallery) | §2.2.21 |
| labelControl (Text Label) | §2.2.25 |
| menu (Menu) | §2.2.28 |
| separator (Separator) | §2.2.34 |
| splitButton (Split Button) | §2.2.38 |
| toggleButton (Toggle Button) | §2.2.43 |



| Attributes | Description |
|--------------------------------|--|
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is</p> |

| Attributes | Description |
|---|--|
| | <p>specified, then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 285 1128 310"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getScreentip</code> (<code>getScreentip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 674 1230 699"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSupertip</code> (<code>getSupertip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1035 1205 1060"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getVisible</code> (<code>getVisible</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1396 1166 1421"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>id</code> (Control ID)</p> | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> |

| Attributes | Description |
|------------------------------------|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1013 275" data-cs="2" data-kind="parent"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 596 821 617" data-cs="2" data-kind="parent"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 940 1305 1283" data-cs="2" data-kind="parent"> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|---|
| | <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> | <p>Specifies the ID of a built-in control which this control should be inserted before. If</p> |

| Attributes | Description |
|---|--|
| (ID of Built-in Control to Insert Before) | <p>the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> |

| Attributes | Description |
|---|--|
| | <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> |

| Attributes | Description |
|------------------------------|---|
| | <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Group">
  <sequence>
    <sequence>
      <choice minOccurs="0" maxOccurs="1000">
        <group ref="EG_Controls"/>
        <element name="separator" type="CT_Separator"/>
      </choice>
    </sequence>
    <element name="dialogBoxLauncher" type="CT_DialogLauncher" minOccurs="0"
      maxOccurs="1"/>
  </sequence>
  <attributeGroup ref="AG_IDAttributes"/>
  <attributeGroup ref="AG_Label"/>
  <attributeGroup ref="AG_Image"/>
  <attributeGroup ref="AG_PositionAttributes"/>
  <attributeGroup ref="AG_Screentip"/>
  <attributeGroup ref="AG_Visible"/>
  <attributeGroup ref="AG_Keytip"/>
</complexType>
```

2.2.24 item (Selection Item)

This element specifies an item in a selection-type control.

[Example: Consider a drop-down control with three selection items, as follows:

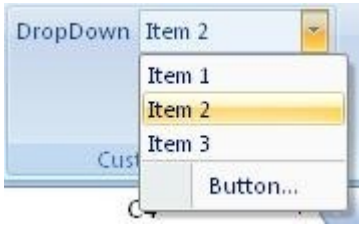


Figure 14: A drop-down control with selection items


This would be specified using the following XML fragment:

```
<dropDown id="dropDown" label="DropDown">
  <item id="item1" label="Item 1" />
  <item id="item2" label="Item 2" />
  <item id="item3" label="Item 3" />
  <button id="button" label="Button..." />
</dropDown>
```

end example]

| Parent Elements |
|---|
| comboBox (§2.2.7); dropDown (§2.2.17); gallery (§2.2.21); gallery (§2.2.22) |

| Attributes | Description |
|-------------------------|---|
| id (Custom Control ID) | <p>Specifies the ID for a custom control. All new custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| image (Custom Image ID) | <p>Specifies the identification information for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified a blank icon should be displayed, unless the <code>showImage</code> or <code>getShowImage</code> attributes are specified and used to hide the icon.</p> |

| Attributes | Description |
|--|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="467 254 980 275" data-cs="2" data-kind="parent"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-In Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified a blank icon should be displayed, unless the <code>showImage</code> or <code>getShowImage</code> attributes are specified and used to hide the icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="467 653 954 674" data-cs="2" data-kind="parent"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST ID</code> simple type (§2.3.5).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified, the label for the control should be blank.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="467 957 1029 978" data-cs="2" data-kind="parent"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST String</code> simple type (§2.3.11).</p> |
| <code>screentip</code> (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="467 1675 1170 1696" data-cs="2" data-kind="parent"><button id="button" imageMso="HappyFace" label="Button"</pre> |

| Attributes | Description |
|------------------------|---|
| | <pre>size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p> <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Item">
  <attribute name="id" type="ST_UniqueID" use="optional"/>
  <attribute name="label" type="ST_String" use="optional"/>
  <attribute name="image" type="ST_Uri" use="optional"/>
  <attribute name="imageMso" type="ST_ID" use="optional"/>
  <attribute name="screentip" type="ST_String" use="optional"/>
  <attribute name="supertip" type="ST_String" use="optional"/>
</complexType>
```

2.2.25 labelControl (Text Label)

This element specifies a control that displays a simple string of text.

[Example: Consider a label control, as follows:

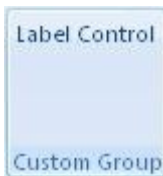


Figure 15: A label control

This would be specified using the following XML fragment:

```
<labelControl id="label" label="Label Control" />
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |


| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> |



| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1127 275" data-cs="2" data-kind="parent"><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 617 1154 638" data-cs="2" data-kind="parent"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 978 1127 999" data-cs="2" data-kind="parent"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1367 1230 1388" data-cs="2" data-kind="parent"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> |


| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1256 275" data-cs="2" data-kind="parent"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 701 1256 722" data-cs="2" data-kind="parent"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSupertip</code> (<code>getSupertip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1089 1208 1110" data-cs="2" data-kind="parent"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getVisible</code> (<code>getVisible</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1451 1167 1472" data-cs="2" data-kind="parent"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>id</code> (Control ID)</p> | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> |

| Attributes | Description |
|--|--|
| | <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 310 1015 338"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 657 824 684"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1003 1307 1346"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>image</code> (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> |

| Attributes | Description |
|--|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1024 275" data-cs="2" data-kind="parent"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 646 1000 667" data-cs="2" data-kind="parent"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1037 1279 1108" data-cs="2" data-kind="parent"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1478 1292 1549" data-cs="2" data-kind="parent"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |

| Attributes | Description |
|---|--|
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---------------------------------|---|
| | <pre data-bbox="516 218 1078 241"><button id="button" label="Custom Button" /></pre> <p data-bbox="487 273 1328 359">This specifies a custom button with a label of <code>Custom Button</code>. end example] The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| screentip (Screentip) | <p data-bbox="487 384 1354 495">Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="487 504 1101 529">[Example: Consider a button with a screentip, as follows:</p>  <p data-bbox="487 877 1107 903">This would be specified using the following XML fragment:</p> <pre data-bbox="516 930 1216 980"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="487 1010 634 1035">end example]</p> <p data-bbox="487 1043 1328 1094">The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| showImage (Show Image) | <p data-bbox="487 1119 1360 1205">Specifies whether this control should display an icon. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="487 1213 1330 1266">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="487 1274 1258 1299">[Example: Consider a button which does not display an icon, as follows:</p>  <p data-bbox="487 1486 1107 1512">This would be specified using the following XML fragment:</p> <pre data-bbox="516 1539 987 1589"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="487 1619 634 1644">end example]</p> <p data-bbox="487 1652 1318 1703">The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

| Attributes | Description |
|-------------------------------|---|
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> |

| Attributes | Description |
|------------------------------|--|
| | The possible values for this attribute are defined by the ST_String simple type (§2.3.11). |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>isVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_LabelControl">
  <complexContent>
    <restriction base="CT_Control">
      <attribute name="image" use="prohibited"/>
      <attribute name="imageMso" use="prohibited"/>
      <attribute name="getImage" use="prohibited"/>
      <attribute name="keytip" use="prohibited"/>
      <attribute name="getKeytip" use="prohibited"/>
      <attribute name="showImage" use="prohibited"/>
      <attribute name="getShowImage" use="prohibited"/>
    </restriction>
  </complexContent>
</complexType>
```


2.2.26 menu (Unsize Menu)

This element specifies a menu control which, due to its location, cannot have its size changed. The `size` attribute is not present. It otherwise behaves identically to the regular `menu` (§2.2.28) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); splitButton (§2.2.38); splitButton (§2.2.36) |

| Child Elements | Subclause |
|-----------------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| checkBox (Check Box) | §2.2.6 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |

| Child Elements | Subclause |
|-------------------------------------|-------------------------|
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Unsize Menu) | §2.2.26 |
| menuSeparator (Menu Separator) | §2.2.30 |
| splitButton (Unsize Split Button) | §2.2.36 |
| toggleButton (Unsize Toggle Button) | §2.2.42 |

| Attributes | Description |
|--|--|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|--|--|
| | <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|--|---|
| | <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> | Specifies the name of a callback function which should be called to determine the |


| Attributes | Description |
|--|--|
| (getSupertip Callback) | <p>supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> |

| Attributes | Description |
|--|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are</p> |

| Attributes | Description |
|---|--|
| | <p>defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 285 1279 359"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 726 1295 800"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1188 1295 1262"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1629 1312 1682"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ...</pre> |

| Attributes | Description |
|--|---|
| | <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <p><code>itemSize</code> (Item Size)</p> | <p>Specifies the size of the child controls in this menu.</p> <p>If this attribute is omitted, the menu's child controls should default to the normal size.</p> <p>[Example: Consider a menu control with large menu items, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><menu id="menu" label="Menu with large items" itemSize="large"> <button id="button1" label="Button 1" imageMso="HappyFace" /> <button id="button2" label="Button 2" imageMso="Paste" /> <button id="button3" label="Button 3" imageMso="Copy" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ItemSize</code> simple type (§2.3.6).</p> |
| <p><code>keytip</code> (Keytip)</p> | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <p><code>label</code> (Label)</p> | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> |

| Attributes | Description |
|---|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1078 275" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 961 1218 1016" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>showImage</code> (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 1570 987 1625" style="background-color: #f0f0f0; padding: 5px;"><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean</p> |

| Attributes | Description |
|------------------------|---|
| | datatype. |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_MenuRegular">
  <complexContent>
    <extension base="CT_ControlBase">
      <sequence>
        <choice minOccurs="0" maxOccurs="1000">
          <group ref="EG_MenuControlsBase"/>
          <group ref="EG_MenuOrSplitButtonRegular"/>
        </choice>
      </sequence>
      <attribute name="itemSize" type="ST_ItemSize" use="optional"/>
      <attributeGroup ref="AG_Description"/>
      <attributeGroup ref="AG_IDAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.27 menu (Menu with Title)

This element specifies a menu control which, due to its location, can optionally include a title string via the title or getTitle attributes. It otherwise behaves identically to the regular menu (§2.2.28) element.

[Example: Consider a menu control with a title, as follows:

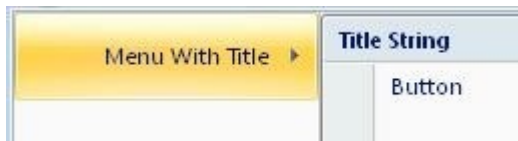


Figure 16: A menu control with title

This would be specified with the following XML fragment:

```
<menu id="menu" label="Menu With Title" title="Title String">
  <button id="button" label="Button" />
</menu>
```

</menu>

end example]

| Parent Elements |
|---|
| menu (§2.2.27); officeMenu (§2.2.31); splitButton (§2.2.37) |

| Child Elements | Subclause |
|---------------------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| checkBox (Check Box) | §2.2.6 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Menu with Title) | §2.2.27 |
| menuSeparator (Menu Separator) | §2.2.30 |
| splitButton (Split Button with Title) | §2.2.37 |
| toggleButton (Unsize Toggle Button) | §2.2.42 |

| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|---|---|
| | <pre data-bbox="516 218 1166 243"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="488 273 1360 323">This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p data-bbox="488 331 1360 382">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 411 354 483"><code>getImage</code> (<code>getImage</code> Callback)</p> | <p data-bbox="488 411 1360 462">Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p data-bbox="488 470 1360 520">The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p data-bbox="488 529 1008 554">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 583 1127 609"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="488 638 1360 688">This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p data-bbox="488 697 1360 747">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 768 354 840"><code>getKeytip</code> (<code>getKeytip</code> Callback)</p> | <p data-bbox="488 768 1360 819">Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p data-bbox="488 827 1360 877">The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p data-bbox="488 886 1008 911">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 940 1153 966"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="488 995 1360 1045">This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p data-bbox="488 1054 1360 1104">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1125 354 1197"><code>getLabel</code> (<code>getLabel</code> Callback)</p> | <p data-bbox="488 1125 1360 1176">Specifies the name of a callback function which should be called to determine the label of this control.</p> <p data-bbox="488 1184 1360 1234">The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p data-bbox="488 1243 1008 1268">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1297 1127 1323"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="488 1352 1360 1402">This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p data-bbox="488 1411 1360 1461">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1482 354 1554"><code>getScreentip</code> (<code>getScreentip</code> Callback)</p> | <p data-bbox="488 1482 1360 1533">Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p data-bbox="488 1541 1360 1625">The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p data-bbox="488 1633 1008 1659">[Example: Consider the following XML fragment:</p> |


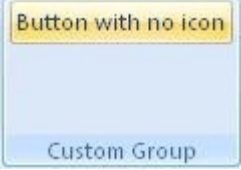
| Attributes | Description |
|---|---|
| | <pre data-bbox="516 218 1230 243"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="488 275 1357 327">This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p data-bbox="488 333 1357 386">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p data-bbox="488 411 1321 464">Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p data-bbox="488 470 1357 522">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="488 529 1330 581">The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p data-bbox="488 588 1008 615">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 642 1256 667"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="488 695 1382 772">This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p data-bbox="488 779 1357 831">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p data-bbox="488 852 1321 905">Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p data-bbox="488 911 1357 963">This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p data-bbox="488 970 1330 1022">The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p data-bbox="488 1029 1008 1056">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1083 1256 1108"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="488 1136 1382 1213">This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p data-bbox="488 1220 1357 1272">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> (<code>getSupertip</code> Callback) | <p data-bbox="488 1293 1357 1346">Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p data-bbox="488 1352 1305 1404">The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p data-bbox="488 1411 1008 1438">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1465 1203 1491"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="488 1518 1382 1575">This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p data-bbox="488 1581 1357 1633">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getTitle</code> (<code>getTitle</code> Callback) | <p data-bbox="488 1656 1357 1709">Specifies the name of a callback function which should be called to determine the title of this control.</p> |


| Attributes | Description |
|--|--|
| | <p>The <code>title</code> and <code>getTitle</code> attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><menu id="menu" label="Menu" getTitle="GetMenuTitle"> ... </menu></pre> <p>This specifies that the <code>GetMenuTitle</code> callback function should be called when the application needs to determine the title of the menu. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these</p> |

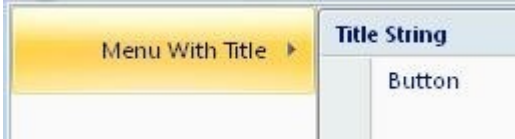
| Attributes | Description |
|---|---|
| | <p>attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <p><code>image</code> (Custom Image ID)</p> | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| <p><code>imageMso</code> (Built-in Image ID)</p> | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <p><code>insertAfterMso</code> (ID of Built-in Control to Insert After)</p> | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the</p> |

| Attributes | Description |
|---|--|
| | <p>controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 310 1284 386"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 751 1295 827"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1220 1295 1295"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1661 1308 1707"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ...</pre> |

| Attributes | Description |
|--|--|
| | <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <p><code>itemSize</code> (Item Size)</p> | <p>Specifies the size of the child controls in this menu.</p> <p>If this attribute is omitted, the menu's child controls should default to the normal size.</p> <p>[Example: Consider a menu control with large menu items, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><menu id="menu" label="Menu with large items" itemSize="large"> <button id="button1" label="Button 1" imageMso="HappyFace" /> <button id="button2" label="Button 2" imageMso="Paste" /> <button id="button3" label="Button 3" imageMso="Copy" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ItemSize</code> simple type (§2.3.6).</p> |
| <p><code>keytip</code> (Keytip)</p> | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <p><code>label</code> (Label)</p> | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> |

| Attributes | Description |
|---|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1078 275" data-cs="2" data-kind="parent"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 968 1214 1016" data-cs="2" data-kind="parent"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>showImage</code> (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 1577 987 1625" data-cs="2" data-kind="parent"><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean</p> |

| Attributes | Description |
|------------------------|---|
| | datatype. |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| title (Title) | <p>Specifies a string which should be displayed as the title of the control.</p> <p>The title and getTitle attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider a menu control with a title, as follows:</p>  <p>This would be specified with the following XML fragment:</p> <pre><menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_MenuWithTitle">
  <complexContent>
    <extension base="CT_ControlBase">
      <sequence>
        <choice minOccurs="0" maxOccurs="1000">
          <group ref="EG_MenuControlsBase"/>
          <group ref="EG_MenuOrSplitButtonWithTitle"/>
        </choice>
      </sequence>
      <attributeGroup ref="AG_IDAttributes"/>
      <attribute name="itemSize" type="ST_ItemSize" use="optional"/>
      <attributeGroup ref="AG_Title"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.28 menu (Menu)

This element specifies a drop-menu control.

[Example: Consider a menu control, as follows:



Figure 17: A menu control


This would be specified using the following XML fragment:

```
<menu id="menu" label="Menu" imageMso="HappyFace" >
  <button id="button1" label="Button 1" imageMso="FileSave" />
  <button id="button2" label="Button 2" imageMso="Bold" />
  <button id="button3" label="Button 3" imageMso="Undo" />
</menu>
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|-------------------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| checkBox (Check Box) | §2.2.6 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Unsize Menu) | §2.2.26 |
| menuSeparator (Menu Separator) | §2.2.30 |
| splitButton (Unsize Split Button) | §2.2.36 |
| toggleButton (Unsize Toggle Button) | §2.2.42 |

| Attributes | Description |
|---|---|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_LongString</code> simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|--|--|
| | <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (<code>getImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getScreentip</code> (<code>getScreentip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> |



| Attributes | Description |
|---|---|
| | <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowImage</code> (<code>getShowImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getShowLabel</code> (<code>getShowLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSize</code> (<code>getSize</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getSupertip</code> (<code>getSupertip</code> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> |



| Attributes | Description |
|---|--|
| Callback) | <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> |


| Attributes | Description |
|--|---|
| | <pre> <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> |

| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before</p> |

| Attributes | Description |
|--|--|
| | <p>the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <p><code>itemSize</code> (Item Size)</p> | <p>Specifies the size of the child controls in this menu.</p> <p>If this attribute is omitted, the menu's child controls should default to the normal size.</p> <p>[Example: Consider a menu control with large menu items, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><menu id="menu" label="Menu with large items" itemSize="large"> <button id="button1" label="Button 1" imageMso="HappyFace" /> <button id="button2" label="Button 2" imageMso="Paste" /> <button id="button3" label="Button 3" imageMso="Copy" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ItemSize</code> simple type (§2.3.6).</p> |
| <p><code>itemSize</code> (Item Size)</p> | <p>Specifies the size of the child controls in this menu.</p> <p>If this attribute is omitted, the menu's child controls should default to the normal size.</p> <p>[Example: Consider a menu control with large menu items, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><menu id="menu" label="Menu with large items" itemSize="large"> <button id="button1" label="Button 1" imageMso="HappyFace" /> <button id="button2" label="Button 2" imageMso="Paste" /></pre> |

| Attributes | Description |
|-----------------------|--|
| | <pre><button id="button3" label="Button 3" imageMso="Copy" /> </menu></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_ItemSize simple type (§2.3.6).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button"</pre> |

| Attributes | Description |
|-------------------------------|---|
| | <pre>size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showLabel (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>size (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>end example]</p> |

| Attributes | Description |
|---|---|
| | <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 254 1052 275" style="background-color: #f0f0f0; padding: 5px;"><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre data-bbox="516 982 1219 1058" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>tag (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1373 1117 1423" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>visible (Control Visibility)</p> | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|------------|--|
| | <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Menu">
  <complexContent>
    <extension base="CT_MenuRegular">
      <attributeGroup ref="AG_SizeAttributes"/>
      <attribute name="itemSize" type="ST_ItemSize" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.29 menu (Dynamic Menu Root XML Element)

This element specifies the root tag of the XML string returned by a dynamic menu control.

[Example: Consider a dynamic menu control, as follows:



Figure 18: A dynamic menu control

This would be specified using the following XML fragment:


```
<dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" />
```

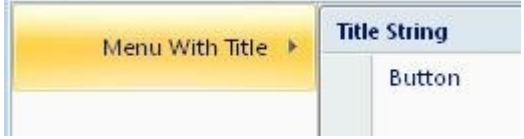
The `GetMenuContent` callback function is called when the menu is dropped, and in this case would return a string with the following XML:

```
<menu xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <button id="button1" label="Button 1" />
  <button id="button2" label="Button 2" />
  <button id="button3" label="Button 3" />
</menu>
```

end example]

| Child Elements | Subclause |
|-------------------------------------|-------------------------|
| button (Unsize Button) | §2.2.3 |
| checkBox (Check Box) | §2.2.6 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Unsize Menu) | §2.2.26 |
| menuSeparator (Menu Separator) | §2.2.30 |
| splitButton (Unsize Split Button) | §2.2.36 |
| toggleButton (Unsize Toggle Button) | §2.2.42 |

| Attributes | Description |
|------------------------------------|---|
| getTitle (getTitle Callback) | <p>Specifies the name of a callback function which should be called to determine the title of this control.</p> <p>The <code>title</code> and <code>getTitle</code> attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><menu id="menu" label="Menu" getTitle="GetMenuTitle"> ... </menu></pre> <p>This specifies that the <code>GetMenuTitle</code> callback function should be called when the application needs to determine the title of the menu. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| itemSize (Item Size) | <p>Specifies the size of the child controls in this menu.</p> <p>If this attribute is not specified, the menu's child controls should default to the normal size.</p> <p>[Example: Consider a menu control with large menu items, as follows:</p>  <p>This would be specified using the following XML fragment:</p> |

| Attributes | Description |
|---------------|---|
| | <pre> <menu id="menu" label="Menu with large items" itemSize="large"> <button id="button1" label="Button 1" imageMso="HappyFace" /> <button id="button2" label="Button 2" imageMso="Paste" /> <button id="button3" label="Button 3" imageMso="Copy" /> </menu> </pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_ItemSize simple type (§2.3.6).</p> |
| title (Title) | <p>Specifies a string which should be displayed as the title of the control.</p> <p>The title and getTitle attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider a menu control with a title, as follows:</p>  <p>This would be specified with the following XML fragment:</p> <pre> <menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu> </pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_MenuRoot">
  <sequence>
    <choice minOccurs="0" maxOccurs="1000">
      <group ref="EG_MenuControlsBase"/>
      <group ref="EG_MenuOrSplitButtonRegular"/>
    </choice>
  </sequence>
  <attributeGroup ref="AG_Title"/>
  <attribute name="itemSize" type="ST_ItemSize" use="optional"/>
</complexType>

```

2.2.30 menuSeparator (Menu Separator)

This element specifies a horizontal separator line in a menu control. Menu separators can optionally have title strings, which should display as headers in the menu.

[Example: Consider a menu with a separator in between two of its items, as follows:



Figure 19: Menu control with separator

This would be specified using the following XML fragment:

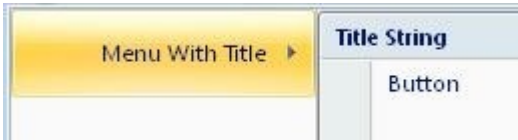
```
<menu id="menu" label="Menu" imageMso="HappyFace" >
  <button id="button1" label="Button 1" imageMso="FileSave" />
  <menuSeparator id="separator" />
  <button id="button2" label="Button 2" imageMso="Bold" />
</menu>
```

end example]

| Parent Elements |
|--|
| menu (§2.2.28); menu (§2.2.26); menu (§2.2.29); menu (§2.2.27); officeMenu (§2.2.31) |

| Attributes | Description |
|------------------------------|---|
| getTitle (getTitle Callback) | <p>Specifies the name of a callback function which should be called to determine the title of this control.</p> <p>The <code>title</code> and <code>getTitle</code> attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><menu id="menu" label="Menu" getTitle="GetMenuTitle"> ... </menu></pre> <p>This specifies that the <code>GetMenuTitle</code> callback function should be called when the application needs to determine the title of the menu. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |

| Attributes | Description |
|--|--|
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case ex is an XML namespace prefix for the namespace http://www.example.com. This XML fragment refers to a tab in that namespace with an ID of OtherTab. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The insertAfterMso, insertAfterQ, insertBeforeMso and insertBeforeQ attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of MyTab should be inserted after the built-in tab with an ID of TabHome. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The insertAfterMso, insertAfterQ, insertBeforeMso and insertBeforeQ attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ...</pre> |

| Attributes | Description |
|---|--|
| | <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>title</code> (Title) | <p>Specifies a string which should be displayed as the title of the control.</p> <p>The <code>title</code> and <code>getTitle</code> attributes are mutually exclusive. If neither attribute is specified no title should be shown.</p> <p>[Example: Consider a menu control with a title, as follows:</p>  <p>This would be specified with the following XML fragment:</p> <pre><menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu></pre> |

| Attributes | Description |
|------------|--|
| | end example] The possible values for this attribute are defined by the ST_String simple type (§2.3.11). |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_MenuSeparator">
  <attributeGroup ref="AG_IDCustom"/>
  <attributeGroup ref="AG_PositionAttributes"/>
  <attributeGroup ref="AG_Title"/>
</complexType>
```

2.2.31 officeMenu (Office Menu)

This element specifies the Office Menu of the application. It is used to reference the built-in Office Menu. This element should not be specified if the containing Custom UI XML document is a Quick Access Toolbar Customizations part.

[Example: Consider the following XML fragment:

```
<officeMenu>
  <control idMso="FileSave" visible="false" />
</officeMenu>
```

This XML fragment specifies that the command with an ID of `FileSave` on the Office Menu should be hidden.

end example]

| Parent Elements |
|------------------|
| ribbon (§2.2.33) |

| Child Elements | Subclause |
|---------------------------------------|-----------|
| button (Unsize Button) | §2.2.3 |
| checkBox (Check Box) | §2.2.6 |
| control (Unsize Control Clone) | §2.2.11 |
| dynamicMenu (Unsize Dynamic Menu) | §2.2.18 |
| gallery (Unsize Gallery) | §2.2.22 |
| menu (Menu with Title) | §2.2.27 |
| menuSeparator (Menu Separator) | §2.2.30 |
| splitButton (Split Button with Title) | §2.2.37 |

| Child Elements | Subclause |
|-------------------------------------|-----------|
| toggleButton (Unsize Toggle Button) | §2.2.42 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OfficeMenu">
  <sequence>
    <choice minOccurs="0" maxOccurs="1000">
      <group ref="EG_MenuControlsBase"/>
      <group ref="EG_MenuOrSplitButtonWithTitle"/>
    </choice>
  </sequence>
</complexType>
```

2.2.32 qat (Quick Access Toolbar)

This element specifies the quick access toolbar. If the containing Custom UI file is a Ribbon Extensibility part the qat element cannot be used unless the startFromScratch attribute on the ribbon element is set to true. In this case only the sharedControls child element should be used. If the containing Custom UI file is a Quick Access Toolbar Customizations part the documentControls child element should be used.

[Example: Consider the following controls on the document-specific quick access toolbar:



Figure 20: Controls on the quick access toolbar

This would be specified using the following XML fragment:

```
<qat>
  <documentControls>
    <control idMso="CalculateNow" />
    <control idMso="HyperlinkInsert" />
  </documentControls>
</qat>
```

end example]

| Parent Elements |
|------------------|
| ribbon (§2.2.33) |

| Child Elements | Subclause |
|--|-----------|
| documentControls (List of Document-Specific Quick Access Toolbar Controls) | §2.2.16 |
| sharedControls (List of Shared Quick Access Toolbar Controls) | §2.2.35 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Qat">
  <sequence>
    <element name="sharedControls" type="CT_QatItems" minOccurs="0"/>
    <element name="documentControls" type="CT_QatItems" minOccurs="0"/>
  </sequence>
</complexType>
```

2.2.33 ribbon (Ribbon)

This element is used to reference the Ribbon of the application and its contents.

| Parent Elements |
|--------------------------------------|
| customUI (§2.2.14) |

| Child Elements | Subclause |
|--|-------------------------|
| contextualTabs (List of Contextual Tab Sets) | §2.2.10 |
| officeMenu (Office Menu) | §2.2.31 |
| qat (Quick Access Toolbar) | §2.2.32 |
| tabs (List of Tabs) | §2.2.40 |

| Attributes | Description |
|---------------------------------------|--|
| startFromScratch (Start From Scratch) | <p>Specifies that the application's built-in ribbon UI should be reduced to a minimal set of features, providing a clean slate on which to build custom UI. If this attribute is omitted its value should default to <i>false</i>. [Example: Consider the following XML fragment:</p> <pre><ribbon startFromScratch="true"> ... </ribbon></pre> <p>This specifies that the application's ribbon should be put into start from scratch mode. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Ribbon">
  <all>
    <element name="officeMenu" type="CT_OfficeMenu" minOccurs="0" maxOccurs="1"/>
    <element name="qat" type="CT_Qat" minOccurs="0" maxOccurs="1">
      <unique name="qatControls">
```

```

<selector xpath="*/*/"/>
<field xpath="@id"/>
</unique>
</element>
<element name="tabs" type="CT_Tabs" minOccurs="0" maxOccurs="1"/>
<element name="contextualTabs" type="CT_ContextualTabs" minOccurs="0" maxOccurs="1"/>
</all>
<attribute name="startFromScratch" type="xsd:boolean" use="optional"/>
</complexType>

```

2.2.34 separator (Separator)

This element specifies a vertical separator line between two sets of controls, either in the Quick Access Toolbar or within `group` elements.

[Example: Consider a vertical separator control between two buttons, as follows:



Figure 21: A vertical separator control

This would be specified using the following XML fragment:

```

<button id="button1" label="Button 1" imageMso="HappyFace" size="large" />
<separator id="separator" />
<button id="button2" label="Button 2" imageMso="HappyFace" size="large" />

```

end example]

| Parent Elements |
|---|
| documentControls (§2.2.16); group (§2.2.23); sharedControls (§2.2.35) |

| Attributes | Description |
|--|--|
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |

| Attributes | Description |
|--|--|
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> |

| Attributes | Description |
|---|---|
| | The possible values for this attribute are defined by the ST_ID simple type (§2.3.5). |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_QID simple type (§2.3.9).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> |

| Attributes | Description |
|------------|--|
| | <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Separator">
  <attributeGroup ref="AG_IDCustom"/>
  <attributeGroup ref="AG_Visible"/>
  <attributeGroup ref="AG_PositionAttributes"/>
</complexType>
```

2.2.35 sharedControls (List of Shared Quick Access Toolbar Controls)

This element specifies the section of the quick access toolbar that is shared between all documents. This element should not be specified if the containing Custom UI XML document is a Quick Access Toolbar Customizations part. If the containing Custom UI XML document is a Ribbon Extensibility part, this element can be used if the `startFromScratch` attribute is set to `true` on the `ribbon` element.

[Example: Consider a Ribbon Extensibility XML document which adds the two buttons to the shared section of the quick access toolbar:



Figure 22: Shared controls on the quick access toolbar

This would be specified using the following XML fragment:

```
<qat>
  <sharedControls>
    <button id="button1" imageMso="HappyFace" />
    <button idMso="Cut" />
  </sharedControls>
</qat>
```

end example]

| Parent Elements |
|-----------------|
| qat (§2.2.32) |

| Child Elements | Subclause |
|--|-------------------------|
| button (Unsize Button) | §2.2.3 |
| control (Quick Access Toolbar Control Clone) | §2.2.13 |
| separator (Separator) | §2.2.34 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_QatItems">
  <sequence>
    <choice minOccurs="0" maxOccurs="1000">
      <element name="control" type="CT_ControlCloneQat"/>
      <element name="button" type="CT_ButtonRegular"/>
      <element name="separator" type="CT_Separator"/>
    </choice>
  </sequence>
</complexType>
```

2.2.36 splitButton (Unsize Split Button)

This element specifies a split button control which, due to its location, cannot have its size changed. The `size` attribute is not present. It otherwise behaves identically to the regular `splitButton` ([§2.2.38](#)) element.

| Parent Elements |
|--|
| buttonGroup (§2.2.5); menu (§2.2.28); menu (§2.2.26); menu (§2.2.29) |

| Child Elements | Subclause |
|---|-------------------------|
| button (Button Inside of a Split Button) | §2.2.4 |
| menu (Unsize Menu) | §2.2.26 |
| toggleButton (Toggle Button Inside of a Split Button) | §2.2.44 |

| Attributes | Description |
|-------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled</p> |



| Attributes | Description |
|--|--|
| | <p>button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <code>getEnabled</code> (<code>getEnabled</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (<code>getImage</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getKeytip</code> (<code>getKeytip</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getLabel</code> (<code>getLabel</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type</p> |



| Attributes | Description |
|--|---|
| | (§2.3.2). |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowImage (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|--|
| | <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"></pre> |

| Attributes | Description |
|--|--|
| | <pre> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre> |

| Attributes | Description |
|---|--|
| | <p></tab></p> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |

| Attributes | Description |
|-----------------------|---|
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show) | <p>Specifies whether this control should display an icon.</p> |

| Attributes | Description |
|------------------------|---|
| Image) | <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>]</p> |

| Attributes | Description |
|------------------------------|--|
| | <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_SplitButtonRegular">
  <complexContent>
    <extension base="CT_SplitButtonRestricted">
      <sequence minOccurs="0">
        <choice minOccurs="0">
          <element name="button" type="CT_VisibleButton"/>
          <element name="toggleButton" type="CT_VisibleToggleButton"/>
        </choice>
          <element name="menu" type="CT_MenuRegular"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

2.2.37 splitButton (Split Button with Title)

This element specifies a split button control which, due to its location, can optionally include a title string via the `title` or `getTitle` attributes. It otherwise behaves identically to the regular `splitButton` (§2.2.38) element.

| Parent Elements |
|--|
| <code>menu</code> (§2.2.27); <code>officeMenu</code> (§2.2.31) |

| Child Elements | Subclause |
|--|-----------|
| <code>button</code> (Button Inside of a Split Button) | §2.2.4 |
| <code>menu</code> (Menu with Title) | §2.2.27 |
| <code>toggleButton</code> (Toggle Button Inside of a Split Button) | §2.2.44 |


| Attributes | Description |
|---|---|
| <code>enabled</code> (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <code>getEnabled</code> (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getImage</code> (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of</p> |



| Attributes | Description |
|---|---|
| | <p>these attributes are specified then no icon should be displayed. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 289 1128 310"><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getKeytip</code> (<code>getKeytip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 653 1154 674"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getLabel</code> (<code>getLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1010 1128 1031"><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getScreentip</code> (<code>getScreentip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1398 1230 1419"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowImage</code> (<code>getShowImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither</p> |


| Attributes | Description |
|---|---|
| | <p>attribute is specified the control should display its icon. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 285 1256 310"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 735 1256 760"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSupertip</code> (<code>getSupertip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control. The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1121 1205 1146"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getVisible</code> (<code>getVisible</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <ul style="list-style-type: none"> ▪ The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible. <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1520 1166 1545"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |

| Attributes | Description |
|-----------------------------|--|
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally</p> |

| Attributes | Description |
|--|--|
| | <p>within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST Uri</code> simple type (§2.3.14).</p> |
| <code>imageMso</code> (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control.</p> <p>The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after</p> |

| Attributes | Description |
|---|--|
| | <p>the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is</p> |

| Attributes | Description |
|--|--|
| | <p>specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example] The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>screentip</code> (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example] The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <p><code>showImage</code> (Show Image)</p> | <p>Specifies whether this control should display an icon. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> |

| Attributes | Description |
|-------------------------------|---|
| | The possible values for this attribute are defined by the XML Schema boolean datatype. |
| showLabel (Show Label) | <p>Specifies whether this control should display its label. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example] The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| supertip (Supertip) | <p>Specifies a string which should be shown as the supertip of the control. The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown. [Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example] The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control. If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> |

| Attributes | Description |
|---|---|
| | <ul style="list-style-type: none"> This specifies a button with a <code>tag</code> value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example] <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_SplitButtonWithTitle">
  <complexContent>
    <extension base="CT_SplitButtonRestricted">
      <sequence minOccurs="0">
        <choice minOccurs="0">
          <element name="button" type="CT_VisibleButton"/>
          <element name="toggleButton" type="CT_VisibleToggleButton"/>
        </choice>
        <element name="menu" type="CT_MenuWithTitle"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

2.2.38 splitButton (Split Button)

This element specifies a split button control. A split button control is composed of either a button or a toggle button, and a drop-down menu. The icon and label shown on the split button come from the `button` or `toggleButton` child element.

[Example: Consider a split button control, as follows:



Figure 23: A split button control

This would be specified using the following XML fragment:

```
<splitButton id="splitButton" size="large" >
  <button id="button" imageMso="HappyFace" label="Split Button" />
  <menu id="menu">
    <button id="button1" label="Button 1" />
    <button id="button2" label="Button 2" />
  </menu>
</splitButton>
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Child Elements | Subclause |
|---|-----------|
| button (Button Inside of a Split Button) | §2.2.4 |
| menu (Unsize Menu) | §2.2.26 |
| toggleButton (Toggle Button Inside of a Split Button) | §2.2.44 |

| Attributes | Description |
|----------------------------------|---|
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> |



| Attributes | Description |
|--|--|
| | The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2). |
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |



| Attributes | Description |
|---|---|
| getShowImage (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSize (getSize Callback) | <p>Specifies the name of a callback function which should be called to determine the size of this control.</p> <p>The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the</p> |


| Attributes | Description |
|----------------------------------|--|
| | <p>application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the IsButtonVisible callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of MyButton. end example]</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of Bold. end example]</p> <p>The possible values for this attribute are defined by the ST_ID simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The id, idQ and idMso attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"></pre> |

| Attributes | Description |
|--|--|
| | <pre> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after</p> |

| Attributes | Description |
|---|--|
| | <p>the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is</p> |

| Attributes | Description |
|-------------------------------|---|
| | <p>specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| <p>label (Label)</p> | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>screentip (Screentip)</p> | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> |

| Attributes | Description |
|--|---|
| | <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>showLabel</code> (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>size</code> (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Size</code> simple type (§2.3.10).</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither</p> |

| Attributes | Description |
|------------------------------|---|
| | <p>attribute is specified no supertip for this control should be shown. [Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control. If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control. The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible. [Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:


```

<complexType name="CT_SplitButton">
  <complexContent>
    <extension base="CT_SplitButtonRegular">
      <attributeGroup ref="AG_SizeAttributes"/>
    </extension>
  </complexContent>
</complexType>

```

2.2.39 tab (Tab)

This element specifies a ribbon tab control.

[Example: Consider the following XML fragment:

```

<tab id="MyTab" label="My Custom Tab">
  ...
</tab>

```

This XML fragment specifies a custom tab with label `My Custom Tab`. end example]


| Parent Elements |
|----------------------------------|
| tabs (§2.2.40); tabSet (§2.2.41) |

| Child Elements | Subclause |
|----------------|-----------|
| group (Group) | §2.2.23 |

| Attributes | Description |
|--------------------------------|--|
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|--|--|
| | <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>getVisible</code> (<code>getVisible</code> Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>id</code> (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>idQ</code> (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"></pre> |

| Attributes | Description |
|---|---|
| | <pre> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a tab in that namespace with an ID of <code>OtherTab</code>. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertAfterMso</code> (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertAfterQ</code> (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>insertBeforeMso</code> (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are</p> |

| Attributes | Description |
|--|--|
| | <p>defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>insertBeforeQ</code> (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| <code>keytip</code> (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| <code>label</code> (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |

| Attributes | Description |
|------------------------------|--|
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible. [Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Tab">
  <sequence>
    <choice minOccurs="0" maxOccurs="100">
      <element name="group" type="CT_Group"/>
    </choice>
  </sequence>
  <attributeGroup ref="AG_IDAttributes"/>
  <attributeGroup ref="AG_Label"/>
  <attributeGroup ref="AG_PositionAttributes"/>
  <attributeGroup ref="AG_Visible"/>
  <attributeGroup ref="AG_Keytip"/>
</complexType>
```

2.2.40 tabs (List of Tabs)

This element specifies a list of ribbon tab controls. This element should not be specified if the containing Custom UI XML document is a Quick Access Toolbar Customizations part.

| Parent Elements |
|------------------|
| ribbon (§2.2.33) |

| Child Elements | Subclause |
|----------------|-------------------------|
| tab (Tab) | §2.2.39 |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Tabs">
  <sequence>
    <element name="tab" type="CT_Tab" minOccurs="1" maxOccurs="100"/>
  </sequence>
</complexType>
```

2.2.41 tabSet (Contextual Tab Set)

This element specifies a contextual tab set control. As the `id` and `idQ` attributes are not present, this element can only be used to refer to existing built-in tab sets. This element cannot be used to create new contextual tab sets.

[Example: Consider the following XML fragment:

```
<tabSet idMso="TabSetPictureTools">
  <tab id="tab" label="Custom Tab">
    ...
  </tab>
</tabSet>
```

This XML fragment would be used to add a new custom tab to the tab set with an ID of `TabSetPictureTools`.

end example]

| Parent Elements |
|--|
| contextualTabs (§2.2.10) |

| Child Elements | Subclause |
|----------------|-------------------------|
| tab (Tab) | §2.2.39 |

| Attributes | Description |
|-------------------------------------|--|
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> |

| Attributes | Description |
|---|---|
| | <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <code>idMso</code> (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This would be used to create a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |


The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_TabSet">
  <sequence>
    <element name="tab" type="CT_Tab" minOccurs="0" maxOccurs="50"/>
  </sequence>
  <attribute name="idMso" type="ST_ID" use="required"/>
  <attributeGroup ref="AG_Visible"/>
</complexType>
```

2.2.42 toggleButton (Unsize Toggle Button)

This element specifies a toggle button control which, due to its location, cannot have its size changed. The `size` attribute is not present. It otherwise behaves identically to the regular `toggleButton` (§2.2.43) element.

| Parent Elements |
|--|
| <code>buttonGroup</code> (§2.2.5); <code>menu</code> (§2.2.28); <code>menu</code> (§2.2.26); <code>menu</code> (§2.2.29); <code>menu</code> (§2.2.27); <code>officeMenu</code> (§2.2.31) |

| Attributes | Description |
|--|---|
| description (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_LongString</code> simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> |


| Attributes | Description |
|---|---|
| | <pre data-bbox="516 218 1166 243"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="488 275 1360 327">This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p data-bbox="488 333 1360 386">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 411 354 485"><code>getImage</code> (<code>getImage</code> Callback)</p> | <p data-bbox="488 411 1360 464">Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p data-bbox="488 470 1360 522">The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p data-bbox="488 529 1008 554">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 585 1127 611"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="488 642 1360 695">This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p data-bbox="488 701 1360 753">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 768 354 842"><code>getKeytip</code> (<code>getKeytip</code> Callback)</p> | <p data-bbox="488 768 1360 821">Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p data-bbox="488 827 1360 879">The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p data-bbox="488 886 1008 911">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 942 1154 968"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="488 999 1360 1052">This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p data-bbox="488 1058 1360 1110">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1131 451 1184"><code>getLabel</code> (<code>getLabel</code> Callback)</p> | <p data-bbox="488 1131 1360 1184">Specifies the name of a callback function which should be called to determine the label of this control.</p> <p data-bbox="488 1190 1360 1243">The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p data-bbox="488 1249 1008 1274">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1306 1127 1331"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="488 1362 1360 1415">This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p data-bbox="488 1421 1360 1474">The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p data-bbox="237 1495 370 1568"><code>getPressed</code> (<code>getPressed</code> Callback)</p> | <p data-bbox="488 1495 1360 1547">Specifies the name of a callback function which should be called to determine the toggled state of this control.</p> <p data-bbox="488 1554 1219 1579">If this attribute is omitted the control should default to the off state.</p> <p data-bbox="488 1585 1008 1610">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1642 1240 1667"><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p data-bbox="488 1698 1360 1724">This specifies that the <code>IsButtonToggled</code> callback function will be called when the</p> |



| Attributes | Description |
|--|---|
| | <p>application needs to determine the toggle state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The getScreentip and screentip attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the GetButtonScreentip callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getShowImage (getShowImage Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control.</p> <p>This attribute should have no effect if the size or getSize attributes specify that the control should be large.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the IsButtonImageVisible callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getShowLabel (getShowLabel Callback) | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control.</p> <p>This attribute should have no effect if the size or getSize attributes specify that the control should be large.</p> <p>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the IsButtonLabelVisible callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> |


| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 254 1206 275"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getVisible</code> (<code>getVisible</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 617 1166 638"><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>id</code> (Control ID)</p> | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1003 1011 1024"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| <p><code>idMso</code> (Built-in Control ID)</p> | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1346 824 1367"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| <p><code>idQ</code> (Qualified Control ID)</p> | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1692 630 1713"><customUI</pre> |

| Attributes | Description |
|--|--|
| | <pre> xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|---|---|
| | <pre data-bbox="516 218 1279 296"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="488 327 1382 380">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p data-bbox="488 390 1382 411">The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p data-bbox="488 443 1382 495">Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="488 506 1382 600">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="488 611 1008 632">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 663 1295 741"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="488 772 1382 825">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p data-bbox="488 835 1382 877">The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertBeforeMso (ID of Built-in Control to Insert Before) | <p data-bbox="488 911 1382 963">Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="488 974 1382 1068">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="488 1079 1008 1100">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1131 1295 1209"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="488 1241 1382 1293">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p data-bbox="488 1304 1382 1325">The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p data-bbox="488 1352 1382 1404">Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p data-bbox="488 1415 1382 1509">The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p data-bbox="488 1520 1008 1541">[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1572 1312 1650"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="488 1682 1382 1734">This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> |

| Attributes | Description |
|------------------------------|--|
| | The possible values for this attribute are defined by the ST_QID simple type (§2.3.9). |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p> |

| Attributes | Description |
|-------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label</p> |

| Attributes | Description |
|--|--|
| | <p>is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p><code>supertip</code> (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p><code>tag</code> (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of 123456, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p><code>visible</code> (Control Visibility)</p> | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end</p> |

| Attributes | Description |
|------------|--|
| | example] The possible values for this attribute are defined by the XML Schema boolean datatype. |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ToggleButtonRegular">
  <complexContent>
    <extension base="CT_ButtonRegular">
      <attribute name="getPressed" type="ST_Delegate" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.43 toggleButton (Toggle Button)

This element specifies a toggle button control which can be toggled between the pressed and unpressed states by the end-user.

[Example: Consider a toggle button control, as follows:



Figure 24: A toggle button control


This would be specified with the following XML fragment:

```
<toggleButton id="toggleButton" label="Toggle Button" />
```

end example]

| Parent Elements |
|-------------------------------|
| box (§2.2.1); group (§2.2.23) |

| Attributes | Description |
|------------------------------|---|
| description (Description) | Specifies a detailed description of the control, which should be displayed in detailed views. The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text. [Example: Consider a button with a detailed description, as follows: |

| Attributes | Description |
|---|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| <p>enabled (Enabled State)</p> | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>getDescription (getDescription Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| <p>getEnabled (getEnabled Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |


| Attributes | Description |
|--|--|
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getPressed (getPressed Callback) | <p>Specifies the name of a callback function which should be called to determine the toggled state of this control.</p> <p>If this attribute is omitted the control should default to the off state.</p> <p>[Example: Consider the following XML fragment:</p> <pre><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p>This specifies that the <code>IsButtonToggled</code> callback function will be called when the application needs to determine the toggle state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither</p> |



| Attributes | Description |
|--|--|
| | <p>attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 310 1230 338"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowImage</code> (<code>getShowImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 730 1256 758"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1178 1256 1205"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSize</code> (<code>getSize</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the size of this control. The <code>getSize</code> and <code>size</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1566 1101 1593"><button id="button" getSize="GetButtonSize" /></pre> <p>This specifies that the <code>GetButtonSize</code> callback function will be called when the application needs to determine the size of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type</p> |



| Attributes | Description |
|--|--|
| | (§2.3.2). |
| getSupertip (getSupertip Callback) | <p>Specifies the name of a callback function which should be called to determine the supertip of this control.</p> <p>The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other</p> |

| Attributes | Description |
|---------------------------------------|--|
| | <p>Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 348 1305 688"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ... </group> </tab> </tabs> </ribbon> </customUI></pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1108 1024 1129"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1507 1000 1528"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> |

| Attributes | Description |
|--|--|
| Control to Insert After) | <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 365 1284 443"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 806 1295 884"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertBeforeMso (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre data-bbox="516 1268 1295 1346"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> |

| Attributes | Description |
|-------------------------------------|---|
| | <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control. The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically. [Example: Consider a button with KeyTip 'K', as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Keytip</code> simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control. The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control. The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider a button with a screentip, as follows:</p> |

| Attributes | Description |
|-------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| showImage (Show Image) | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| showLabel (Show Label) | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label</p> |

| Attributes | Description |
|----------------------------|---|
| | <p>is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>size (Control Size)</p> | <p>Specifies the size of the control.</p> <p>The <code>size</code> and <code>getSize</code> attributes are mutually exclusive. If neither attribute is specified, the control's size should default to the normal size.</p> <p>[Example: Consider a large button, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Size simple type (§2.3.10).</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>tag (Tag)</p> | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string.</p> |

| Attributes | Description |
|---|---|
| | <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a <code>tag</code> value of <code>123456</code>, which will be passed to the <code>ButtonClicked</code> callback function. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.3.11).</p> |
| <code>visible</code> (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of <code>TabHome</code> should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:


```
<complexType name="CT_ToggleButton">
  <complexContent>
    <extension base="CT_ToggleButtonRegular">
      <attributeGroup ref="AG_SizeAttributes"/>
    </extension>
  </complexContent>
</complexType>
```

2.2.44 toggleButton (Toggle Button Inside of a Split Button)

This element specifies a toggle button control that is part of a split button control. The `visible` and `getVisible` attributes are not present because the visibility is controlled by the split button. This element otherwise behaves like the regular `toggleButton` ([§2.2.43](#)) element.

| Parent Elements |
|--|
| <code>splitButton</code> (§2.2.38); <code>splitButton</code> (§2.2.36); <code>splitButton</code> (§2.2.37) |

| Attributes | Description |
|--|--|
| <code>description</code> (Description) | <p>Specifies a detailed description of the control, which should be displayed in detailed views.</p> <p>The <code>description</code> and <code>getDescription</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider a button with a detailed description, as follows:</p> |

| Attributes | Description |
|--|---|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_LongString simple type (§2.3.8).</p> |
| enabled (Enabled State) | <p>Specifies the enabled state of the control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control which would be otherwise disabled by the application.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button which would always be disabled. A permanently disabled button is not very useful, thus the <code>enabled</code> attribute is not commonly used. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| getDescription (getDescription Callback) | <p>Specifies the name of a callback function which should be called to determine the detailed description of this control.</p> <p>The <code>getDescription</code> and <code>description</code> attributes are mutually exclusive. If neither attribute is specified, the control should not display any detailed text.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>This specifies that the <code>GetButtonDescription</code> callback function should be called when the application needs to determine the detailed description of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| getEnabled (getEnabled Callback) | <p>Specifies the name of a callback function which should be called to determine the enabled state of this control.</p> <p>The <code>getEnabled</code> and <code>enabled</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being enabled.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>This specifies that the <code>IsButtonEnabled</code> callback function will be called when the application needs to determine the enabled state of the button. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |



| Attributes | Description |
|--|--|
| getImage (getImage Callback) | <p>Specifies the name of a callback function which should be called to determine the icon of this control.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>This specifies that the <code>GetButtonImage</code> callback function will be called when the application needs to determine the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getKeytip (getKeytip Callback) | <p>Specifies the name of a callback function which should be called to determine the suggested KeyTip of this control.</p> <p>The <code>getKeytip</code> and <code>keytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>This specifies that the <code>GetButtonKeytip</code> callback function will be called when the application needs to determine the KeyTip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getLabel (getLabel Callback) | <p>Specifies the name of a callback function which should be called to determine the label of this control.</p> <p>The <code>getLabel</code> and <code>label</code> attributes are mutually exclusive. If neither attribute is specified, then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>This specifies that the <code>GetButtonLabel</code> callback function will be called when the application needs to determine the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getPressed (getPressed Callback) | <p>Specifies the name of a callback function which should be called to determine the toggled state of this control.</p> <p>If this attribute is omitted the control should default to the off state.</p> <p>[Example: Consider the following XML fragment:</p> <pre><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p>This specifies that the <code>IsButtonToggled</code> callback function will be called when the application needs to determine the toggle state of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| getScreentip (getScreentip Callback) | <p>Specifies the name of a callback function which should be called to determine the screentip of this control.</p> <p>The <code>getScreentip</code> and <code>screentip</code> attributes are mutually exclusive. If neither</p> |


| Attributes | Description |
|--|--|
| | <p>attribute is specified the application should display the label of the control as the screentip or display no screentip at all. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 310 1230 338"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>This specifies that the <code>GetButtonScreentip</code> callback function will be called when the application needs to determine the screentip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowImage</code> (<code>getShowImage</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the icon of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 730 1256 758"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>This specifies that the <code>IsButtonImageVisible</code> callback function will be called when the application needs to determine whether to display the icon of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getShowLabel</code> (<code>getShowLabel</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine whether the application should display the label of this control. This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large. The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1178 1256 1205"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>This specifies that the <code>IsButtonLabelVisible</code> callback function will be called when the application needs to determine whether to display the label of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| <p><code>getSupertip</code> (<code>getSupertip</code> Callback)</p> | <p>Specifies the name of a callback function which should be called to determine the supertip of this control. The <code>getSupertip</code> and <code>supertip</code> attributes are mutually exclusive. If neither attribute is specified, no supertip for this control should be shown. [Example: Consider the following XML fragment:</p> <pre data-bbox="516 1566 1203 1593"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>This specifies that the <code>GetButtonSupertip</code> callback function will be called when the application needs to determine the supertip of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type</p> |


| Attributes | Description |
|----------------------------------|--|
| | (§2.3.2). |
| getVisible (getVisible Callback) | <p>Specifies the name of a callback function which should be called to determine the visibility state of this control.</p> <p>The <code>getVisible</code> and <code>visible</code> attributes are mutually exclusive. If neither attribute is specified, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>This specifies that the <code>IsButtonVisible</code> callback function will be called when the application needs to determine the visibility of the button. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Delegate</code> simple type (§2.3.2).</p> |
| id (Control ID) | <p>Specifies the ID for a custom control. All custom controls must have unique IDs. The ID of a control should be passed to callback functions in order to identify which control corresponds to the function call.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an ID of <code>MyButton</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_UniqueID</code> simple type (§2.3.13).</p> |
| idMso (Built-in Control ID) | <p>Specifies the ID of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| idQ (Qualified Control ID) | <p>Specifies a qualified ID for a control.</p> <p>The <code>idQ</code> attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p>The <code>id</code>, <code>idQ</code> and <code>idMso</code> attributes are mutually exclusive. At least one of these attributes shall be specified.</p> <p>[Example: Consider the following XML fragment:</p> <pre><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <tab idQ="ex:OtherTab" label="Shared Tab"> <group id="MyGroup" label="My Group"> ...</pre> |

| Attributes | Description |
|--|--|
| | <pre> </group> </tab> </tabs> </ribbon> </customUI> </pre> <p>In this case <code>ex</code> is an XML namespace prefix for the namespace <code>http://www.example.com</code>. This XML fragment refers to a <code>tab</code> in that namespace with an ID of <code>OtherTab</code>. If that <code>tab</code> cannot be found, it will be created. A new group belonging to this file will be added to the <code>tab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| image (Custom Image ID) | <p>Specifies the relationship ID for an image which should be used as the icon for this control. This attribute is used to specify an embedded picture that resides locally within the containing file.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon should be the embedded image file referenced by the relationship ID of <code>ForestPic</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_Uri</code> simple type (§2.3.14).</p> |
| imageMso (Built-in Image ID) | <p>Specifies the ID of a built-in image which should be used as the icon of this control. The contents of this attribute are application-defined and should be ignored if not understood.</p> <p>The <code>getImage</code>, <code>image</code> and <code>imageMso</code> attributes are mutually exclusive. If none of these attributes are specified then no icon should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that should use the built-in image with an ID of <code>Bold</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertAfterMso (ID of Built-in Control to Insert After) | <p>Specifies the ID of a built-in control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |

| Attributes | Description |
|---|--|
| insertAfterQ (Qualified ID of Control to Insert After) | <p>Specifies the qualified ID of a control which this control should be inserted after. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted after the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| insertBeforeMso (ID of Built-in Control to Insert Before) | <p>Specifies the ID of a built-in control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the built-in tab with an ID of <code>TabHome</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_ID</code> simple type (§2.3.5).</p> |
| insertBeforeQ (Qualified ID of Control to Insert Before) | <p>Specifies the qualified ID of a control which this control should be inserted before. If the value of this attribute is not understood it should be ignored.</p> <p>The <code>insertAfterMso</code>, <code>insertAfterQ</code>, <code>insertBeforeMso</code> and <code>insertBeforeQ</code> attributes are mutually exclusive. If none of these attributes are specified, the controls should be appended to the existing set of controls, in the order they are defined in the XML.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>This specifies that a new custom tab with an ID of <code>MyTab</code> should be inserted before the custom tab with a qualified ID of <code>x:OtherTab</code>. end example]</p> <p>The possible values for this attribute are defined by the <code>ST_QID</code> simple type (§2.3.9).</p> |
| keytip (Keytip) | <p>Specifies a string which should be used as the suggested KeyTip for this control.</p> <p>The <code>keytip</code> and <code>getKeytip</code> attributes are mutually exclusive. If neither attribute is specified, the application should generate a KeyTip for the control automatically.</p> <p>[Example: Consider a button with KeyTip 'K', as follows:</p> |

| Attributes | Description |
|------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_Keytip simple type (§2.3.7).</p> |
| label (Label) | <p>Specifies a string which should be used as the label for this control.</p> <p>The <code>label</code> and <code>getLabel</code> attributes are mutually exclusive. If neither attribute is specified then no label should be displayed.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of <code>Custom Button</code>. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| onAction (onAction Callback) | <p>Specifies the name of a callback function which should be called when this control is invoked by the user.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button which should call the <code>ButtonClicked</code> callback function when it is invoked. end example]</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type (§2.3.2).</p> |
| screentip (Screentip) | <p>Specifies a string which should be shown as the screentip for this control.</p> <p>The <code>screentip</code> and <code>getScreentip</code> attributes are mutually exclusive. If neither attribute is specified the application should display the label of the control as the screentip or display no screentip at all.</p> <p>[Example: Consider a button with a screentip, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button"</pre> |

| Attributes | Description |
|-------------------------------|---|
| | <pre>size="large" screentip="This is the screentip" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| <p>showImage (Show Image)</p> | <p>Specifies whether this control should display an icon.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showImage</code> and <code>getShowImage</code> attributes are mutually exclusive. If neither attribute is specified the control should display its icon.</p> <p>[Example: Consider a button which does not display an icon, as follows:</p>  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>showLabel (Show Label)</p> | <p>Specifies whether this control should display its label.</p> <p>This attribute should have no effect if the <code>size</code> or <code>getSize</code> attributes specify that the control should be large.</p> <p>The <code>showLabel</code> and <code>getShowLabel</code> attributes are mutually exclusive. If neither attribute is specified the control should default to showing its label.</p> <p>[Example: Consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button which has a label but does not show it. Even though the label is hidden, it will be provided to accessibility tools. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |
| <p>supertip (Supertip)</p> | <p>Specifies a string which should be shown as the supertip of the control.</p> <p>The <code>supertip</code> and <code>getSupertip</code> attributes are mutually exclusive. If neither attribute is specified no supertip for this control should be shown.</p> <p>[Example: Consider a control with a supertip, as follows:</p> |

| Attributes | Description |
|------------------------------|--|
| |  <p>This would be specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| tag (Tag) | <p>Specifies an arbitrary string that can be used to hold data or identify the control. The contents of this attribute should be passed to any callback functions specified on this control.</p> <p>If this attribute is omitted the control's tag value should default to an empty string. [Example: Consider the following XML fragment:</p> <pre><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of 123456, which will be passed to the ButtonClicked callback function. end example]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.3.11).</p> |
| visible (Control Visibility) | <p>Specifies the visibility state of the control.</p> <p>The isVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control should default to being visible.</p> <p>[Example: Consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>This specifies that the built-in tab with an ID of TabHome should be hidden. end example]</p> <p>The possible values for this attribute are defined by the XML Schema boolean datatype.</p> |

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_VisibleToggleButton">
  <complexContent>
    <restriction base="CT_ToggleButtonRegular">
```

```

    <attribute name="visible" use="prohibited"/>
    <attribute name="getVisible" use="prohibited"/>
  </restriction>
</complexContent>
</complexType>

```

2.3 Simple Types

This is the complete list of simple types in the <http://schemas.microsoft.com/office/2006/01/customui> namespace.

2.3.1 ST_BoxStyle (Box Style)

Specifies the layout style of a `box` control.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

| Enumeration Value | Description |
|-------------------------|--|
| horizontal (Horizontal) | Specifies that the child controls should be laid out horizontally. |
| vertical (Vertical) | Specifies that the child controls should be laid out vertically. |

| Referenced By |
|--|
| <code>box@boxStyle</code> (§ 2.2.1) |

The following XML Schema fragment defines the contents of this simple type:

```

<simpleType name="ST_BoxStyle">
  <restriction base="xsd:string">
    <enumeration value="horizontal"/>
    <enumeration value="vertical"/>
  </restriction>
</simpleType>

```

2.3.2 ST_Delegate (Callback Function Name)

Specifies the name of a callback function. The format of this string shall be application-defined and should be ignored if not understood.

[Example: Macro scripts, add-in callback functions, and so on. end example]

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

Referenced By

box@getVisible (§2.2.1); button@getDescription (§2.2.2); button@getDescription (§2.2.3); button@getDescription (§2.2.4); button@getEnabled (§2.2.2); button@getEnabled (§2.2.3); button@getEnabled (§2.2.4); button@getImage (§2.2.2); button@getImage (§2.2.3); button@getImage (§2.2.4); button@getKeytip (§2.2.2); button@getKeytip (§2.2.3); button@getKeytip (§2.2.4); button@getLabel (§2.2.2); button@getLabel (§2.2.3); button@getLabel (§2.2.4); button@getScreentip (§2.2.2); button@getScreentip (§2.2.3); button@getScreentip (§2.2.4); button@getShowImage (§2.2.2); button@getShowImage (§2.2.3); button@getShowImage (§2.2.4); button@getShowLabel (§2.2.2); button@getShowLabel (§2.2.3); button@getShowLabel (§2.2.4); button@getSize (§2.2.2); button@getSupertip (§2.2.2); button@getSupertip (§2.2.3); button@getSupertip (§2.2.4); button@getVisible (§2.2.2); button@getVisible (§2.2.3); button@getVisible (§2.2.4); button@onAction (§2.2.2); button@onAction (§2.2.3); button@onAction (§2.2.4); buttonGroup@getVisible (§2.2.5); checkBox@getDescription (§2.2.6); checkBox@getEnabled (§2.2.6); checkBox@getImage (§2.2.6); checkBox@getKeytip (§2.2.6); checkBox@getLabel (§2.2.6); checkBox@getPressed (§2.2.6); checkBox@getScreentip (§2.2.6); checkBox@getShowImage (§2.2.6); checkBox@getShowLabel (§2.2.6); checkBox@getSupertip (§2.2.6); checkBox@getVisible (§2.2.6); checkBox@onAction (§2.2.6); comboBox@getEnabled (§2.2.7); comboBox@getImage (§2.2.7); comboBox@getItemCount (§2.2.7); comboBox@getItemID (§2.2.7); comboBox@getItemImage (§2.2.7); comboBox@getItemLabel (§2.2.7); comboBox@getItemScreentip (§2.2.7); comboBox@getItemSupertip (§2.2.7); comboBox@getKeytip (§2.2.7); comboBox@getLabel (§2.2.7); comboBox@getScreentip (§2.2.7); comboBox@getShowImage (§2.2.7); comboBox@getShowLabel (§2.2.7); comboBox@getSupertip (§2.2.7); comboBox@getText (§2.2.7); comboBox@getVisible (§2.2.7); comboBox@onChange (§2.2.7); command@getEnabled (§2.2.8); command@onAction (§2.2.8); control@getDescription (§2.2.12); control@getDescription (§2.2.13); control@getEnabled (§2.2.12); control@getEnabled (§2.2.13); control@getEnabled (§2.2.11); control@getImage (§2.2.12); control@getImage (§2.2.13); control@getImage (§2.2.11); control@getKeytip (§2.2.12); control@getKeytip (§2.2.13); control@getKeytip (§2.2.11); control@getLabel (§2.2.12); control@getLabel (§2.2.13); control@getLabel (§2.2.11); control@getScreentip (§2.2.12); control@getScreentip (§2.2.13); control@getScreentip (§2.2.11); control@getShowImage (§2.2.12); control@getShowImage (§2.2.13); control@getShowImage (§2.2.11); control@getShowLabel (§2.2.12); control@getShowLabel (§2.2.13); control@getShowLabel (§2.2.11); control@getSize (§2.2.12); control@getSize (§2.2.13); control@getSupertip (§2.2.12); control@getSupertip (§2.2.13); control@getSupertip (§2.2.11); control@getVisible (§2.2.12); control@getVisible (§2.2.13); control@getVisible (§2.2.11); control@onAction (§2.2.12); customUI@loadImage (§2.2.14); customUI@onLoad (§2.2.14); dropDown@getEnabled (§2.2.17); dropDown@getImage (§2.2.17); dropDown@getItemCount (§2.2.17); dropDown@getItemID (§2.2.17); dropDown@getItemImage (§2.2.17); dropDown@getItemLabel (§2.2.17); dropDown@getItemScreentip (§2.2.17); dropDown@getItemSupertip (§2.2.17); dropDown@getKeytip (§2.2.17); dropDown@getLabel (§2.2.17); dropDown@getScreentip (§2.2.17); dropDown@getSelectedItemID (§2.2.17); dropDown@getSelectedItemIndex (§2.2.17); dropDown@getShowImage (§2.2.17); dropDown@getShowLabel (§2.2.17); dropDown@getSupertip (§2.2.17); dropDown@getVisible (§2.2.17); dropDown@onAction (§2.2.17); dynamicMenu@getContent (§2.2.19); dynamicMenu@getContent (§2.2.18); dynamicMenu@getDescription (§2.2.19); dynamicMenu@getDescription (§2.2.18); dynamicMenu@getEnabled (§2.2.19); dynamicMenu@getEnabled (§2.2.18); dynamicMenu@getImage (§2.2.19); dynamicMenu@getImage (§2.2.18); dynamicMenu@getKeytip (§2.2.19); dynamicMenu@getKeytip (§2.2.18); dynamicMenu@getLabel (§2.2.19); dynamicMenu@getLabel (§2.2.18); dynamicMenu@getScreentip (§2.2.19); dynamicMenu@getScreentip (§2.2.18); dynamicMenu@getShowImage (§2.2.19); dynamicMenu@getShowImage (§2.2.18); dynamicMenu@getShowLabel (§2.2.19); dynamicMenu@getShowLabel (§2.2.18); dynamicMenu@getSize (§2.2.19); dynamicMenu@getSupertip (§2.2.19); dynamicMenu@getSupertip (§2.2.18); dynamicMenu@getVisible (§2.2.19); dynamicMenu@getVisible (§2.2.18); editBox@getEnabled (§2.2.20); editBox@getImage (§2.2.20); editBox@getKeytip (§2.2.20); editBox@getLabel (§2.2.20); editBox@getScreentip (§2.2.20); editBox@getShowImage (§2.2.20); editBox@getShowLabel (§2.2.20); editBox@getSupertip (§2.2.20); editBox@getText (§2.2.20); editBox@getVisible (§2.2.20); editBox@onChange (§2.2.20); gallery@getDescription (§2.2.21); gallery@getDescription (§2.2.22); gallery@getEnabled (§2.2.21); gallery@getEnabled (§2.2.22); gallery@getImage (§2.2.21); gallery@getImage (§2.2.22); gallery@getItemCount (§2.2.21); gallery@getItemCount (§2.2.22); gallery@getItemHeight (§2.2.21); gallery@getItemHeight (§2.2.22); gallery@getItemID (§2.2.21); gallery@getItemID (§2.2.22); gallery@getItemImage (§2.2.21); gallery@getItemImage (§2.2.22);

Referenced By

gallery@getItemLabel (§2.2.21); gallery@getItemLabel (§2.2.22); gallery@getItemScreentip (§2.2.21); gallery@getItemScreentip (§2.2.22); gallery@getItemSupertip (§2.2.21); gallery@getItemSupertip (§2.2.22); gallery@getItemWidth (§2.2.21); gallery@getItemWidth (§2.2.22); gallery@getKeytip (§2.2.21); gallery@getKeytip (§2.2.22); gallery@getLabel (§2.2.21); gallery@getLabel (§2.2.22); gallery@getScreentip (§2.2.21); gallery@getScreentip (§2.2.22); gallery@getSelectedItemID (§2.2.21); gallery@getSelectedItemID (§2.2.22); gallery@getSelectedItemIndex (§2.2.21); gallery@getSelectedItemIndex (§2.2.22); gallery@getShowImage (§2.2.21); gallery@getShowImage (§2.2.22); gallery@getShowLabel (§2.2.21); gallery@getShowLabel (§2.2.22); gallery@getSize (§2.2.21); gallery@getSupertip (§2.2.21); gallery@getSupertip (§2.2.22); gallery@getVisible (§2.2.21); gallery@getVisible (§2.2.22); gallery@onAction (§2.2.21); gallery@onAction (§2.2.22); group@getImage (§2.2.23); group@getImage (§2.2.23); group@getScreentip (§2.2.23); group@getSupertip (§2.2.23); group@getVisible (§2.2.23); labelControl@getEnabled (§2.2.25); labelControl@getImage (§2.2.25); labelControl@getKeytip (§2.2.25); labelControl@getLabel (§2.2.25); labelControl@getScreentip (§2.2.25); labelControl@getShowImage (§2.2.25); labelControl@getShowLabel (§2.2.25); labelControl@getSupertip (§2.2.25); labelControl@getVisible (§2.2.25); menu@getDescription (§2.2.28); menu@getDescription (§2.2.28); menu@getEnabled (§2.2.28); menu@getEnabled (§2.2.27); menu@getImage (§2.2.28); menu@getImage (§2.2.26); menu@getImage (§2.2.27); menu@getKeytip (§2.2.28); menu@getKeytip (§2.2.26); menu@getKeytip (§2.2.27); menu@getLabel (§2.2.28); menu@getLabel (§2.2.26); menu@getLabel (§2.2.27); menu@getScreentip (§2.2.28); menu@getScreentip (§2.2.26); menu@getScreentip (§2.2.27); menu@getShowImage (§2.2.28); menu@getShowImage (§2.2.26); menu@getShowImage (§2.2.27); menu@getShowLabel (§2.2.28); menu@getShowLabel (§2.2.26); menu@getShowLabel (§2.2.27); menu@getSize (§2.2.28); menu@getSupertip (§2.2.28); menu@getSupertip (§2.2.26); menu@getSupertip (§2.2.27); menu@getTitle (§2.2.29); menu@getTitle (§2.2.27); menu@getVisible (§2.2.28); menu@getVisible (§2.2.26); menu@getVisible (§2.2.27); menuSeparator@getTitle (§2.2.30); separator@getVisible (§2.2.34); splitButton@getEnabled (§2.2.38); splitButton@getEnabled (§2.2.36); splitButton@getEnabled (§2.2.37); splitButton@getImage (§2.2.38); splitButton@getImage (§2.2.36); splitButton@getImage (§2.2.37); splitButton@getKeytip (§2.2.38); splitButton@getKeytip (§2.2.36); splitButton@getKeytip (§2.2.37); splitButton@getLabel (§2.2.38); splitButton@getLabel (§2.2.36); splitButton@getLabel (§2.2.37); splitButton@getScreentip (§2.2.38); splitButton@getScreentip (§2.2.36); splitButton@getScreentip (§2.2.37); splitButton@getShowImage (§2.2.38); splitButton@getShowImage (§2.2.36); splitButton@getShowImage (§2.2.37); splitButton@getShowLabel (§2.2.38); splitButton@getShowLabel (§2.2.36); splitButton@getShowLabel (§2.2.37); splitButton@getSize (§2.2.38); splitButton@getSupertip (§2.2.38); splitButton@getSupertip (§2.2.36); splitButton@getSupertip (§2.2.37); splitButton@getVisible (§2.2.38); splitButton@getVisible (§2.2.36); splitButton@getVisible (§2.2.37); tab@getKeytip (§2.2.39); tab@getLabel (§2.2.39); tab@getVisible (§2.2.39); tabSet@getVisible (§2.2.41); toggleButton@getDescription (§2.2.43); toggleButton@getDescription (§2.2.42); toggleButton@getDescription (§2.2.44); toggleButton@getEnabled (§2.2.43); toggleButton@getEnabled (§2.2.42); toggleButton@getEnabled (§2.2.44); toggleButton@getImage (§2.2.43); toggleButton@getImage (§2.2.42); toggleButton@getImage (§2.2.44); toggleButton@getKeytip (§2.2.43); toggleButton@getKeytip (§2.2.42); toggleButton@getKeytip (§2.2.44); toggleButton@getLabel (§2.2.43); toggleButton@getLabel (§2.2.42); toggleButton@getLabel (§2.2.44); toggleButton@getPressed (§2.2.43); toggleButton@getPressed (§2.2.42); toggleButton@getPressed (§2.2.44); toggleButton@getScreentip (§2.2.43); toggleButton@getScreentip (§2.2.42); toggleButton@getScreentip (§2.2.44); toggleButton@getShowImage (§2.2.43); toggleButton@getShowImage (§2.2.42); toggleButton@getShowImage (§2.2.44); toggleButton@getShowLabel (§2.2.43); toggleButton@getShowLabel (§2.2.42); toggleButton@getShowLabel (§2.2.44); toggleButton@getSize (§2.2.43); toggleButton@getSupertip (§2.2.43); toggleButton@getSupertip (§2.2.42); toggleButton@getSupertip (§2.2.44); toggleButton@getVisible (§2.2.43); toggleButton@getVisible (§2.2.42); toggleButton@getVisible (§2.2.44); toggleButton@onAction (§2.2.43); toggleButton@onAction (§2.2.42); toggleButton@onAction (§2.2.44)

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_Delegate">
  <restriction base="xsd:string">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

2.3.3 ST_GalleryItemWidthHeight (Gallery Item Width or Height)

Specifies the width or height of gallery items, in pixels.

This simple type's contents are a restriction of the XML Schema positiveInteger datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 4096.

| Referenced By |
|--|
| gallery@itemHeight (§2.2.21); gallery@itemHeight (§2.2.22); gallery@itemWidth (§2.2.21); gallery@itemWidth (§2.2.22) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_GalleryItemWidthHeight">
  <restriction base="xsd:positiveInteger">
    <minInclusive value="1"/>
    <maxInclusive value="4096"/>
  </restriction>
</simpleType>
```

2.3.4 ST_GalleryRowColumnCount (Gallery Row or Column Count)

Specifies the count of rows or columns in a gallery control.

This simple type's contents are a restriction of the XML Schema positiveInteger datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 1024.

| Referenced By |
|--|
| gallery@columns (§2.2.21); gallery@columns (§2.2.22); gallery@rows (§2.2.21); gallery@rows (§2.2.22) |

The following XML Schema fragment defines the contents of this simple type:


```

<simpleType name="ST_GalleryRowColumnCount">
  <restriction base="xsd:positiveInteger">
    <minInclusive value="1"/>
    <maxInclusive value="1024"/>
  </restriction>
</simpleType>

```

2.3.5 ST_ID (Control ID)

Specifies the ID of a built-in control. The format of this string is defined by per application by the Custom UI Control ID Tables ([Appendix A: Custom UI Control ID Tables](#)).

This simple type's contents are a restriction of the XML Schema NCName datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

| Referenced By |
|---|
| <p> box@insertAfterMso (§2.2.1); box@insertBeforeMso (§2.2.1); button@idMso (§2.2.2); button@idMso (§2.2.3); button@idMso (§2.2.4); button@imageMso (§2.2.2); button@imageMso (§2.2.3); button@imageMso (§2.2.4); button@insertAfterMso (§2.2.2); button@insertAfterMso (§2.2.3); button@insertAfterMso (§2.2.4); button@insertBeforeMso (§2.2.2); button@insertBeforeMso (§2.2.3); button@insertBeforeMso (§2.2.4); buttonGroup@insertAfterMso (§2.2.5); buttonGroup@insertBeforeMso (§2.2.5); checkBox@idMso (§2.2.6); checkBox@imageMso (§2.2.6); checkBox@insertAfterMso (§2.2.6); checkBox@insertBeforeMso (§2.2.6); comboBox@idMso (§2.2.7); comboBox@imageMso (§2.2.7); comboBox@insertAfterMso (§2.2.7); comboBox@insertBeforeMso (§2.2.7); command@idMso (§2.2.8); control@id (§2.2.13); control@idMso (§2.2.12); control@idMso (§2.2.13); control@idMso (§2.2.11); control@imageMso (§2.2.12); control@imageMso (§2.2.13); control@imageMso (§2.2.11); control@insertAfterMso (§2.2.12); control@insertAfterMso (§2.2.13); control@insertAfterMso (§2.2.11); control@insertBeforeMso (§2.2.12); control@insertBeforeMso (§2.2.13); control@insertBeforeMso (§2.2.11); dropDown@idMso (§2.2.17); dropDown@imageMso (§2.2.17); dropDown@insertAfterMso (§2.2.17); dropDown@insertBeforeMso (§2.2.17); dynamicMenu@idMso (§2.2.19); dynamicMenu@idMso (§2.2.18); dynamicMenu@imageMso (§2.2.19); dynamicMenu@imageMso (§2.2.18); dynamicMenu@insertAfterMso (§2.2.19); dynamicMenu@insertAfterMso (§2.2.18); dynamicMenu@insertBeforeMso (§2.2.19); dynamicMenu@insertBeforeMso (§2.2.18); editBox@idMso (§2.2.20); editBox@imageMso (§2.2.20); editBox@insertAfterMso (§2.2.20); editBox@insertBeforeMso (§2.2.20); gallery@idMso (§2.2.21); gallery@idMso (§2.2.22); gallery@imageMso (§2.2.21); gallery@imageMso (§2.2.22); gallery@insertAfterMso (§2.2.21); gallery@insertAfterMso (§2.2.22); gallery@insertBeforeMso (§2.2.21); gallery@insertBeforeMso (§2.2.22); group@idMso (§2.2.23); group@imageMso (§2.2.23); group@insertAfterMso (§2.2.23); group@insertBeforeMso (§2.2.23); item@imageMso (§2.2.24); labelControl@idMso (§2.2.25); labelControl@imageMso (§2.2.25); labelControl@insertAfterMso (§2.2.25); labelControl@insertBeforeMso (§2.2.25); menu@idMso (§2.2.28); menu@idMso (§2.2.26); menu@idMso (§2.2.27); menu@imageMso (§2.2.28); menu@imageMso (§2.2.26); menu@imageMso (§2.2.27); menu@insertAfterMso (§2.2.28); menu@insertAfterMso (§2.2.26); menu@insertAfterMso (§2.2.27); menu@insertBeforeMso (§2.2.28); menu@insertBeforeMso (§2.2.26); menu@insertBeforeMso (§2.2.27); menuSeparator@insertAfterMso (§2.2.30); menuSeparator@insertBeforeMso (§2.2.30); separator@insertAfterMso (§2.2.34); separator@insertBeforeMso (§2.2.34); splitButton@idMso (§2.2.38); splitButton@idMso (§2.2.36); splitButton@idMso (§2.2.37); splitButton@imageMso (§2.2.38); splitButton@imageMso (§2.2.36); splitButton@imageMso (§2.2.37); splitButton@insertAfterMso (§2.2.38); splitButton@insertAfterMso (§2.2.36); splitButton@insertAfterMso (§2.2.37); splitButton@insertBeforeMso (§2.2.38); splitButton@insertBeforeMso (§2.2.36); splitButton@insertBeforeMso (§2.2.37); tab@idMso (§2.2.39); tab@insertAfterMso (§2.2.39); tab@insertBeforeMso (§2.2.39); tabSet@idMso (§2.2.41); </p> |

| Referenced By |
|--|
| toggleButton@idMso (§2.2.43); toggleButton@idMso (§2.2.42); toggleButton@idMso (§2.2.44); toggleButton@imageMso (§2.2.43); toggleButton@imageMso (§2.2.42); toggleButton@imageMso (§2.2.44); toggleButton@insertAfterMso (§2.2.43); toggleButton@insertAfterMso (§2.2.42); toggleButton@insertAfterMso (§2.2.44); toggleButton@insertBeforeMso (§2.2.43); toggleButton@insertBeforeMso (§2.2.42); toggleButton@insertBeforeMso (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_ID">
  <restriction base="xsd:NCName">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

2.3.6 ST_ItemSize (Menu Item Size)

Specifies the size of the child controls in a `menu` control.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

| Enumeration Value | Description |
|-------------------|---|
| large (Large) | Specifies that the child controls should have large sizes. |
| normal (Normal) | Specifies that the child controls should have normal sizes. |

| Referenced By |
|--|
| menu@itemSize (§2.2.28); menu@itemSize (§2.2.26); menu@itemSize (§2.2.29); menu@itemSize (§2.2.27) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_ItemSize">
  <restriction base="xsd:string">
    <enumeration value="normal"/>
    <enumeration value="large"/>
  </restriction>
</simpleType>
```

2.3.7 ST_Keytip (Keytip)

Specifies a KeyTip string.

This simple type's contents are a restriction of the XML Schema token datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 3 characters.

| Referenced By |
|---|
| button@keytip (§2.2.2); button@keytip (§2.2.3); button@keytip (§2.2.4); checkBox@keytip (§2.2.6); comboBox@keytip (§2.2.7); control@keytip (§2.2.12); control@keytip (§2.2.13); control@keytip (§2.2.11); dropDown@keytip (§2.2.17); dynamicMenu@keytip (§2.2.19); dynamicMenu@keytip (§2.2.18); editBox@keytip (§2.2.20); gallery@keytip (§2.2.21); gallery@keytip (§2.2.22); group@keytip (§2.2.23); labelControl@keytip (§2.2.25); menu@keytip (§2.2.28); menu@keytip (§2.2.26); menu@keytip (§2.2.27); splitButton@keytip (§2.2.38); splitButton@keytip (§2.2.36); splitButton@keytip (§2.2.37); tab@keytip (§2.2.39); toggleButton@keytip (§2.2.43); toggleButton@keytip (§2.2.42); toggleButton@keytip (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:

```

<simpleType name="ST_Keytip">
  <restriction base="xsd:token">
    <minLength value="1"/>
    <maxLength value="3"/>
    <whiteSpace value="collapse"/>
  </restriction>
</simpleType>

```

2.3.8 ST_LongString (Long String)

Specifies a string which can have an extended length.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 4096 characters.

| Referenced By |
|---|
| button@description (§2.2.2); button@description (§2.2.3); button@description (§2.2.4); checkBox@description (§2.2.6); control@description (§2.2.12); control@description (§2.2.13); dynamicMenu@description (§2.2.19); dynamicMenu@description (§2.2.18); gallery@description (§2.2.21); gallery@description (§2.2.22); menu@description (§2.2.28); menu@description (§2.2.26); toggleButton@description (§2.2.43); toggleButton@description (§2.2.42); toggleButton@description (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:

```

<simpleType name="ST_LongString">
  <restriction base="xsd:string">
    <minLength value="1"/>
    <maxLength value="4096"/>
  </restriction>
</simpleType>

```

2.3.9 ST_QID (Qualified Control ID)

Specifies a control ID which is qualified by an XML namespace prefix. The prefix determines which namespace the control belongs to.

If the namespace is equal to the Custom UI namespace, the qualified ID will reference the application's built-in control set.

[Example: Consider the following XML fragment:

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
  xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon>
    <tabs>
      <tab idQ="mso:TabHome" visible="false" />
    </tabs>
  </ribbon>
</customUI>
```

In this example the mso namespace prefix is set to the Custom UI namespace, so names qualified with mso will refer to built-in controls. Thus the use of the idQ attribute on the tab element is equivalent to using the idMso attribute, as follows:

```
<tab idMso="TabHome" visible="false" />
```

end example]

If the prefix is set to any other value, qualified IDs will reference controls in a unique custom namespace. If multiple Custom UI documents refer to controls in the same namespace, they can share common containers.

[Example: Consider the following XML fragment:

```
<customUI
  xmlns="http://schemas.microsoft.com/office/2006/01/customui"
  xmlns:ex="http://www.example.com">
  <ribbon>
    <tabs>
      <tab idQ="ex:OtherTab" label="Shared Tab">
        <group id="MyGroup" label="My Group">
          ...
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

In this case ex is an XML namespace prefix for the namespace `http://www.example.com`. This XML fragment refers to a tab in that namespace with an ID of `OtherTab`. If that tab cannot be found, it will be created. A new group belonging to this file will be added to the tab. end example]

This simple type's contents are a restriction of the XML Schema QName datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

Referenced By

box@idQ (§2.2.1); box@insertAfterQ (§2.2.1); box@insertBeforeQ (§2.2.1); button@idQ (§2.2.2); button@idQ (§2.2.3); button@idQ (§2.2.4); button@insertAfterQ (§2.2.2); button@insertAfterQ (§2.2.3); button@insertAfterQ (§2.2.4); button@insertBeforeQ (§2.2.2); button@insertBeforeQ (§2.2.3); button@insertBeforeQ (§2.2.4); buttonGroup@idQ (§2.2.5); buttonGroup@insertAfterQ (§2.2.5); buttonGroup@insertBeforeQ (§2.2.5); checkBox@idQ (§2.2.6); checkBox@insertAfterQ (§2.2.6); checkBox@insertBeforeQ (§2.2.6); comboBox@idQ (§2.2.7); comboBox@insertAfterQ (§2.2.7); comboBox@insertBeforeQ (§2.2.7); control@idQ (§2.2.12); control@idQ (§2.2.13); control@idQ (§2.2.11); control@insertAfterQ (§2.2.12); control@insertAfterQ (§2.2.13); control@insertAfterQ (§2.2.11); control@insertBeforeQ (§2.2.12); control@insertBeforeQ (§2.2.13); control@insertBeforeQ (§2.2.11); dropDown@idQ (§2.2.17); dropDown@insertAfterQ (§2.2.17); dropDown@insertBeforeQ (§2.2.17); dynamicMenu@idQ (§2.2.19); dynamicMenu@idQ (§2.2.18); dynamicMenu@insertAfterQ (§2.2.19); dynamicMenu@insertAfterQ (§2.2.18); dynamicMenu@insertBeforeQ (§2.2.19); dynamicMenu@insertBeforeQ (§2.2.18); editBox@idQ (§2.2.20); editBox@insertAfterQ (§2.2.20); editBox@insertBeforeQ (§2.2.20); gallery@idQ (§2.2.21); gallery@idQ (§2.2.22); gallery@insertAfterQ (§2.2.21); gallery@insertAfterQ (§2.2.22); gallery@insertBeforeQ (§2.2.21); gallery@insertBeforeQ (§2.2.22); group@idQ (§2.2.23); group@insertAfterQ (§2.2.23); group@insertBeforeQ (§2.2.23); labelControl@idQ (§2.2.25); labelControl@insertAfterQ (§2.2.25); labelControl@insertBeforeQ (§2.2.25); menu@idQ (§2.2.28); menu@idQ (§2.2.26); menu@idQ (§2.2.27); menu@insertAfterQ (§2.2.28); menu@insertAfterQ (§2.2.26); menu@insertAfterQ (§2.2.27); menu@insertBeforeQ (§2.2.28); menu@insertBeforeQ (§2.2.26); menu@insertBeforeQ (§2.2.27); menuSeparator@idQ (§2.2.30); menuSeparator@insertAfterQ (§2.2.30); menuSeparator@insertBeforeQ (§2.2.30); separator@idQ (§2.2.34); separator@insertAfterQ (§2.2.34); separator@insertBeforeQ (§2.2.34); splitButton@idQ (§2.2.38); splitButton@idQ (§2.2.36); splitButton@idQ (§2.2.37); splitButton@insertAfterQ (§2.2.38); splitButton@insertAfterQ (§2.2.36); splitButton@insertAfterQ (§2.2.37); splitButton@insertBeforeQ (§2.2.38); splitButton@insertBeforeQ (§2.2.36); splitButton@insertBeforeQ (§2.2.37); tab@idQ (§2.2.39); tab@insertAfterQ (§2.2.39); tab@insertBeforeQ (§2.2.39); toggleButton@idQ (§2.2.43); toggleButton@idQ (§2.2.42); toggleButton@idQ (§2.2.44); toggleButton@insertAfterQ (§2.2.43); toggleButton@insertAfterQ (§2.2.42); toggleButton@insertAfterQ (§2.2.44); toggleButton@insertBeforeQ (§2.2.43); toggleButton@insertBeforeQ (§2.2.42); toggleButton@insertBeforeQ (§2.2.44)

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_QID">
  <restriction base="xsd:QName">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

2.3.10 ST_Size (Control Size)

Specifies the size of a control.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

| Enumeration Value | Description |
|------------------------------|------------------------------------|
| large (Large Control Size) | Specifies the large control size. |
| normal (Normal Control Size) | Specifies the normal control size. |

| Referenced By |
|--|
| button@size (§2.2.2); control@size (§2.2.12); control@size (§2.2.13); dynamicMenu@size (§2.2.19); gallery@size (§2.2.21); menu@size (§2.2.28); splitButton@size (§2.2.38); toggleButton@size (§2.2.43) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_Size">
  <restriction base="xsd:string">
    <enumeration value="normal"/>
    <enumeration value="large"/>
  </restriction>
</simpleType>
```

2.3.11 ST_String (Short String)

Specifies a string with a limited length.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

| Referenced By |
|--|
| button@label (§2.2.2); button@label (§2.2.3); button@label (§2.2.4); button@screentip (§2.2.2); button@screentip (§2.2.3); button@screentip (§2.2.4); button@supertip (§2.2.2); button@supertip (§2.2.3); button@supertip (§2.2.4); button@tag (§2.2.2); button@tag (§2.2.3); button@tag (§2.2.4); checkBox@label (§2.2.6); checkBox@screentip (§2.2.6); checkBox@supertip (§2.2.6); checkBox@tag (§2.2.6); comboBox@label (§2.2.7); comboBox@screentip (§2.2.7); comboBox@sizeString (§2.2.7); comboBox@supertip (§2.2.7); comboBox@tag (§2.2.7); control@label (§2.2.12); control@label (§2.2.13); control@label (§2.2.11); control@screentip (§2.2.12); control@screentip (§2.2.13); control@screentip (§2.2.11); control@supertip (§2.2.12); control@supertip (§2.2.13); control@supertip (§2.2.11); control@tag (§2.2.12); control@tag (§2.2.11); dropDown@label (§2.2.17); dropDown@screentip (§2.2.17); dropDown@sizeString (§2.2.17); dropDown@supertip (§2.2.17); dropDown@tag (§2.2.17); dynamicMenu@label (§2.2.19); dynamicMenu@label (§2.2.18); dynamicMenu@screentip (§2.2.19); dynamicMenu@screentip (§2.2.18); dynamicMenu@supertip (§2.2.19); dynamicMenu@supertip (§2.2.18); dynamicMenu@tag (§2.2.19); dynamicMenu@tag (§2.2.18); editBox@label (§2.2.20); editBox@screentip (§2.2.20); editBox@sizeString (§2.2.20); editBox@supertip (§2.2.20); editBox@tag (§2.2.20); gallery@label (§2.2.21); gallery@label (§2.2.22); gallery@screentip (§2.2.21); gallery@screentip (§2.2.22); gallery@sizeString (§2.2.21); gallery@sizeString (§2.2.22); gallery@supertip (§2.2.21); gallery@supertip (§2.2.22); gallery@tag (§2.2.21); gallery@tag (§2.2.22); group@label (§2.2.23); group@screentip (§2.2.23); group@supertip (§2.2.23); group@tag (§2.2.23); item@label (§2.2.24); item@screentip (§2.2.24); |

| Referenced By |
|---|
| item@supertip (§2.2.24) ; labelControl@label (§2.2.25) ; labelControl@screentip (§2.2.25) ; labelControl@supertip (§2.2.25) ; labelControl@tag (§2.2.25) ; menu@label (§2.2.28) ; menu@label (§2.2.26) ; menu@label (§2.2.27) ; menu@screentip (§2.2.28) ; menu@screentip (§2.2.26) ; menu@screentip (§2.2.27) ; menu@supertip (§2.2.28) ; menu@supertip (§2.2.26) ; menu@supertip (§2.2.27) ; menu@tag (§2.2.28) ; menu@tag (§2.2.26) ; menu@tag (§2.2.27) ; menu@title (§2.2.29) ; menu@title (§2.2.27) ; menuSeparator@title (§2.2.30) ; splitButton@label (§2.2.38) ; splitButton@label (§2.2.36) ; splitButton@label (§2.2.37) ; splitButton@screentip (§2.2.38) ; splitButton@screentip (§2.2.36) ; splitButton@screentip (§2.2.37) ; splitButton@supertip (§2.2.38) ; splitButton@supertip (§2.2.36) ; splitButton@supertip (§2.2.37) ; splitButton@tag (§2.2.38) ; splitButton@tag (§2.2.36) ; splitButton@tag (§2.2.37) ; tab@label (§2.2.39) ; tab@tag (§2.2.39) ; toggleButton@label (§2.2.43) ; toggleButton@label (§2.2.42) ; toggleButton@label (§2.2.44) ; toggleButton@screentip (§2.2.43) ; toggleButton@screentip (§2.2.42) ; toggleButton@screentip (§2.2.44) ; toggleButton@supertip (§2.2.43) ; toggleButton@supertip (§2.2.42) ; toggleButton@supertip (§2.2.44) ; toggleButton@tag (§2.2.43) ; toggleButton@tag (§2.2.42) ; toggleButton@tag (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_String">
  <restriction base="xsd:string">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

2.3.12 ST_StringLength (String Length)

Specifies the length of a string, in characters.

This simple type's contents are a restriction of the XML Schema `positiveInteger` datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 1024.

| Referenced By |
|---|
| comboBox@maxLength (§2.2.7) ; editBox@maxLength (§2.2.20) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_StringLength">
  <restriction base="xsd:positiveInteger">
    <minInclusive value="1"/>
    <maxInclusive value="1024"/>
  </restriction>
</simpleType>
```

2.3.13 ST_UniqueID (Custom Control ID)

Specifies a custom control ID.

This simple type's contents are a restriction of the XML Schema ID datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

| Referenced By |
|---|
| box@id (§2.2.1); button@id (§2.2.2); button@id (§2.2.3); button@id (§2.2.4); buttonGroup@id (§2.2.5); checkBox@id (§2.2.6); comboBox@id (§2.2.7); control@id (§2.2.12); control@id (§2.2.11); dropDown@id (§2.2.17); dynamicMenu@id (§2.2.19); dynamicMenu@id (§2.2.18); editBox@id (§2.2.20); gallery@id (§2.2.21); gallery@id (§2.2.22); group@id (§2.2.23); item@id (§2.2.24); labelControl@id (§2.2.25); menu@id (§2.2.28); menu@id (§2.2.26); menu@id (§2.2.27); menuSeparator@id (§2.2.30); separator@id (§2.2.34); splitButton@id (§2.2.38); splitButton@id (§2.2.36); splitButton@id (§2.2.37); tab@id (§2.2.39); toggleButton@id (§2.2.43); toggleButton@id (§2.2.42); toggleButton@id (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:

```
<simpleType name="ST_UniqueID">
  <restriction base="xsd:ID">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

2.3.14 ST Uri (Image Relationship ID)

Specifies the relationship ID of a part which is the target of a relationship from the containing Custom UI document.

The target part shall be of the Image Part type ("[ECMA-376](#) Part 1 §15.2.13; Image Part").

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a minimum length of 1 characters.
- This simple type's contents have a maximum length of 1024 characters.

| Referenced By |
|---|
| button@image (§2.2.2); button@image (§2.2.3); button@image (§2.2.4); checkBox@image (§2.2.6); comboBox@image (§2.2.7); control@image (§2.2.12); control@image (§2.2.13); control@image (§2.2.11); dropDown@image (§2.2.17); dynamicMenu@image (§2.2.19); dynamicMenu@image (§2.2.18); editBox@image (§2.2.20); gallery@image (§2.2.21); gallery@image (§2.2.22); group@image (§2.2.23); item@image (§2.2.24); labelControl@image (§2.2.25); menu@image (§2.2.28); menu@image (§2.2.26); menu@image (§2.2.27); splitButton@image (§2.2.38); splitButton@image (§2.2.36); splitButton@image (§2.2.37); toggleButton@image (§2.2.43); toggleButton@image (§2.2.42); toggleButton@image (§2.2.44) |

The following XML Schema fragment defines the contents of this simple type:


```
<simpleType name="ST_Uri">
  <restriction base="xsd:string">
    <minLength value="1"/>
    <maxLength value="1024"/>
  </restriction>
</simpleType>
```

3 Appendix A: Custom UI Control ID Tables

3.1 idMso Tables

3.1.1 Word 2007

| idMso | Control Type | Label |
|----------------------------------|--------------|--------------------|
| Spelling | button | Spelling... |
| FileSave | button | Save |
| FilePrint | button | Print |
| ZoomOnePage | button | One Page |
| ZoomPageWidth | button | Page Width |
| Zoom100 | button | Zoom 100% |
| ColumnsDialog | button | More Columns... |
| Numbering | toggleButton | Numbering |
| Bullets | toggleButton | Bullets |
| PageOrientationPortraitLandscape | button | Portrait/Landscape |
| OutdentClassic | button | Decrease Indent |
| IndentClassic | button | Increase Indent |
| DrawingInsert | button | Insert Drawing |
| ChartInsert | button | Chart... |
| FileNew | button | New |
| Copy | button | Copy |
| Cut | button | Cut |
| Paste | button | Paste |
| FileOpen | button | Open |
| EnvelopesAndLabelsDialog | button | Envelopes... |
| Superscript | toggleButton | Superscript |
| Subscript | toggleButton | Subscript |
| UnderlineDouble | toggleButton | Double Underline |
| UnderlineWords | button | Word Underline |
| FontSizeIncreaseWord | button | Grow Font |

| idMso | Control Type | Label |
|------------------------------|---------------------|---------------------|
| FontSizeDecreaseWord | button | Shrink Font |
| FileClose | button | Close |
| TableAutoFormat | button | Table AutoFormat... |
| FormatPainter | toggleButton | Format Painter |
| FilePrintPreview | toggleButton | Print Preview |
| PasteApplyStyle | button | Apply Style |
| Bold | toggleButton | Bold |
| Italic | toggleButton | Italic |
| Underline | toggleButton | Underline |
| ParagraphMarks | toggleButton | Show All |
| AlignLeft | toggleButton | Align Left |
| AlignRight | toggleButton | Align Right |
| AlignCenter | toggleButton | Center |
| AlignJustify | toggleButton | Justify |
| HeaderFooterPageNumberInsert | menu | Page Number |
| Undo | gallery | Undo |
| Redo | gallery | Redo |
| OutlinePromote | button | Promote |
| OutlineDemote | button | Demote |
| OutlineMoveUp | button | Move Up |
| OutlineMoveDown | button | Move Down |
| OutlineDemoteToBodyText | button | Demote to Body Text |
| OutlineExpand | button | Expand |
| OutlineCollapse | button | Collapse |
| TextBoxInsert | button | Text Box |
| FileFind | button | Find File... |
| FindDialog | button | Find... |
| TableExcelSpreadsheetInsert | button | Excel Spreadsheet |
| AutoFormat | button | AutoFormat... |

| idMso | Control Type | Label |
|-----------------------------|---------------------|------------------------------|
| BorderInside | toggleButton | Inside Borders |
| BorderOutside | toggleButton | Outside Borders |
| BorderNone | toggleButton | No Border |
| MailMergeGoToFirstRecord | button | First |
| MailMergeGoToPreviousRecord | button | Previous |
| MailMergeGoToNextRecord | button | Next |
| MailMergeGoToLastRecord | button | Last |
| MailMergeMergeToDocument | button | Edit Individual Documents... |
| MailMergeMergeToPrinter | button | Print Documents... |
| MailMergeAutoCheckForErrors | button | Auto Check for Errors... |
| DataFormSource | button | Data Form |
| MailMergeResultsPreview | toggleButton | Preview Results |
| ObjectsGroup | button | Group |
| ObjectsUngroup | button | Ungroup |
| ObjectBringToFront | button | Bring to Front |
| ObjectSendToBack | button | Send to Back |
| ObjectBringForward | button | Bring Forward |
| ObjectSendBackward | button | Send Backward |
| Magnifier | checkBox | Magnifier |
| PrintPreviewShrinkOnePage | button | Shrink One Page |
| ViewFullScreenView | button | Full Screen |
| VoiceInsert | button | Voice Comment |
| ObjectsSelect | toggleButton | Select Objects |
| TableFind | button | Find |
| MacroRecord | button | Record Macro... |
| MacroRecorderPause | button | Pause Recording |
| MacroPlay | button | Macros |
| ShapeFreeform | toggleButton | Freeform |
| ObjectEditPoints | toggleButton | Edit Points |

| idMso | Control Type | Label |
|---------------------------------|---------------------|---------------------|
| CalloutOptions | button | Callout Options |
| DataFormAddRecord | button | Add |
| DataFormDeleteRecord | button | Delete |
| FieldsUpdate | button | Update |
| DatabaseInsert | button | Insert Database |
| GridSettings | button | Grid Settings... |
| WordPicture | button | Word Picture |
| FormControlEditBox | button | Edit Box |
| FormControlCheckBox | button | Check Box |
| FormControlComboBox | button | Combo Box |
| PropertySheet | button | Property Sheet |
| FieldShading | toggleButton | Show Field Shading |
| ViewDraftView | toggleButton | Draft |
| Lock | toggleButton | Lock |
| AutoSum | button | Sum |
| MasterDocumentShow | toggleButton | Show Document |
| MasterDocumentCreateSubdocument | button | Create |
| MasterDocumentUnlinkSubdocument | button | Unlink |
| MasterDocumentInsertSubdocument | button | Insert... |
| MasterDocumentSplitSubdocuments | button | Split |
| MasterDocumentMergeSubdocuments | button | Merge |
| MasterDocumentLockSubdocument | toggleButton | Lock Document |
| HeaderOrFooterShow | button | Show Header/Footer |
| HeaderFooterPreviousSection | button | Previous Section |
| HeaderFooterNextSection | button | Next Section |
| AlignDialog | button | Align |
| MailMergeDocument | button | Mail Merge Document |
| MergeOptions | button | Merge... |
| MailMergeHelper | button | Mail Merge... |

| idMso | Control Type | Label |
|----------------------------|---------------------|-----------------------------|
| PageSetupDialog | button | Page Setup... |
| BodyTextHide | button | Hide Body Text |
| HeaderFooterLinkToPrevious | toggleButton | Link to Previous |
| OutlineShowFirstLineOnly | checkBox | Show First Line Only |
| OutlineShowTextFormatting | checkBox | Show Text Formatting |
| FontDialog | button | Font... |
| StylesDialogClassic | button | Edit Cell Styles |
| FootnoteInsert | button | Insert Footnote |
| MicrosoftExcel | button | Microsoft Excel |
| MicrosoftAccess | button | Microsoft Access |
| MicrosoftPowerPoint | button | Microsoft Office PowerPoint |
| MicrosoftPublisher | button | Microsoft Publisher |
| MicrosoftProject | button | Microsoft Project |
| ViewPrintLayoutView | toggleButton | Print Layout |
| FieldCodes | toggleButton | View Field Codes |
| DropCapOptionsDialog | button | Drop Cap Options... |
| Strikethrough | toggleButton | Strikethrough |
| TextSmallCaps | toggleButton | Small Caps |
| CellsDelete | button | Delete Cells... |
| TableRowsDelete | button | Delete Rows |
| TableColumnsDelete | button | Delete Columns |
| CellsInsertDialog | button | Insert Cells... |
| TableRowsInsertWord | button | Insert Rows |
| WindowsArrangeAll | button | Arrange All |
| MarginsAdjust | button | Adjust Margins |
| ViewGridlinesWord | checkBox | View Gridlines |
| SubdocumentOpen | button | Open Subdocument |
| WindowSplit | button | Split |
| WindowNew | button | New Window |

| idMso | Control Type | Label |
|----------------------------------|---------------------|------------------------------|
| ReviewAcceptOrRejectChangeDialog | button | Accept/Reject Changes |
| TextAllCaps | toggleButton | All Caps |
| PictureDisassemble | button | Disassemble Picture |
| ChangeCaseDialogClassic | button | Change Case... |
| FontSizeDecrease1Point | button | Shrink Font 1 Pt |
| FontSizeIncrease1Point | button | Grow Font 1 Pt |
| Repaginate | button | Repaginate |
| ReplaceDialog | button | Replace... |
| StartOfLine | button | Start of Line |
| EndOfLine | button | End of Line |
| PagePrevious | button | Previous Page |
| PageNext | button | Next Page |
| StartOfDocument | button | Start of Document |
| EndOfDocument | button | End of Document |
| Grammar | button | Grammar... |
| FileCloseOrCloseAll | button | Close |
| TextToOrFromTable | button | Text to/from Table |
| TableRowsOrColumnsOrCellsInsert | button | Insert Table |
| TableRowsOrColumnsOrCellsDelete | button | Delete Rows/Columns/Cells |
| RedoOrRepeat | button | Redo |
| ProtectOrUnprotectDocument | button | Protect Document |
| FrameInsertOrFormat | button | Insert Frame |
| ObjectsRegroup | button | Regroup |
| AutoFormatChange | button | Tip Wizard 6 |
| AddressBook | button | Address Book... |
| Reply | button | Reply |
| ReplyAll | button | Reply to All |
| Forward | button | Forward |
| MailMove | button | Move Mail |

| idMso | Control Type | Label |
|-----------------------------|---------------------|-----------------------------|
| MailDelete | button | Delete Mail |
| MessagePrevious | button | Previous Item |
| MessageNext | button | Next Item |
| MailSelectNames | button | Select Names... |
| AsianLayoutCharacterScaling | menu | Character Scaling |
| ShapeScribble | toggleButton | Scribble |
| PrintSetupDialog | button | Print Setup... |
| RowHeight | button | Row Height... |
| ColumnWidth | button | Column Width... |
| OleObjectctInsert | button | Object... |
| Cancel | button | Cancel |
| FindNext | button | Find Next |
| PasteDuplicate | button | Duplicate |
| ClipArtInsert | toggleButton | Clip Art... |
| ParagraphSpacingIncrease | button | Increase Paragraph Spacing |
| ParagraphSpacingDecrease | button | Decrease Paragraph Spacing |
| OrganizationChartInsert | button | Organization Chart |
| CombineCharacters | toggleButton | Yoko-Gumi |
| DoubleStrikethrough | toggleButton | Double Strikethrough |
| PictureCrop | toggleButton | Crop |
| ViewOutlineView | toggleButton | Outline |
| FileCloseAll | button | Close All |
| FileSaveAs | button | Save As |
| SaveAll | button | Save All |
| AdvancedFileProperties | button | View Document Properties... |
| DocumentTemplate | button | Document Template |
| CopyAsPicture | button | Copy as Picture... |
| PasteSpecialDialog | button | Paste Special... |

| idMso | Control Type | Label |
|----------------------------------|---------------------|---------------------------|
| SelectAll | button | Select All |
| GoTo | button | Go To... |
| BookmarkInsert | button | Bookmark... |
| FileLinksToFiles | button | Edit Links to Files |
| ViewOnlineLayoutViewClassic | button | Online Layout |
| FootnotesEndnotesShow | button | Show Notes |
| BreakInsertDialog | button | Break |
| DateAndTimeInsert | button | Date & Time... |
| NumberInsert | button | Number... |
| FieldInsert | button | Field... |
| FormField | button | Form Field... |
| CaptionInsert | button | Insert Caption... |
| CrossReferenceInsert | button | Cross-reference... |
| IndexAndTables | button | Index and Tables |
| TextFromFileInsert | button | Text from File... |
| ParagraphDialog | button | Paragraph... |
| TabsDialog | button | Tabs... |
| BordersShadingDialog | button | Borders and Shading... |
| TextDirectionOptionsDialog | button | Text Direction Options... |
| BulletsAndNumberingBulletsDialog | button | Bullets and Numbering... |
| StyleGalleryDialog | button | Style Gallery... |
| FrameDialog | button | Frame... |
| SetLanguage | button | Set Language... |
| WordCount | button | Word Count... |
| AutoCorrect | button | AutoCorrect Options... |
| EnvelopesAndLabels | button | Envelope & Label Wizard |
| LabelsDialog | button | Labels... |
| MergeCells | button | Merge Cells |
| SplitCells | button | Split Cells... |

| idMso | Control Type | Label |
|-----------------------|---------------------|--------------------------------------|
| TableRowSelect | button | Select Row |
| TableColumnSelect | button | Select Column |
| TableSelect | button | Select Table |
| TableCellHeightWidth | button | Cell Height and Width... |
| TableRepeatHeaderRows | toggleButton | Repeat Header Rows |
| ConvertTextToTable | button | Convert Text to Table... |
| TableFormulaDialog | button | Formula... |
| TableSplitTable | button | Split Table |
| ShowClipboard | button | Office Clipboard... |
| NumberingSkip | button | Skip Numbering |
| KeyboardCustomization | button | Customize Keyboard... |
| ShowAllHeadings | button | All |
| ImeDictionaryUpdate | button | Update IME Dictionary... |
| OutlookTaskCreate | button | Create Microsoft Office Outlook Task |
| WindowMinimize | button | Minimize |
| WindowRestore | button | Restore |
| WindowClose | button | Close |
| WindowMove | button | Move |
| WindowSize | button | Size |
| WindowNext | button | Next Window |
| ClearFormats | button | Clear Formats |
| OK | button | OK |
| ClosePane | button | Close |
| PrintPreviewClose | button | Close Print Preview |
| HeaderFooterClose | button | Close Header and Footer |
| ZoomDialog | button | Zoom... |
| About | button | About |
| SortDialogClassic | button | Sort... |
| ConvertTableToText | button | Convert to Text... |

| idMso | Control Type | Label |
|--|---------------------|------------------------|
| ExchangeFolder | button | Exchange Folder... |
| ChartEditDataSource | button | Select Data... |
| WindowMoreWindowsDialog | toggleButton | More Windows... |
| ObjectEditDialog | button | Object... |
| ObjectFormatDialog | button | Object... |
| AutoTextCreate | button | Create AutoText... |
| ContentsAndIndex | button | Contents and Index |
| Help | button | Help |
| FontColorMoreColorsDialog | button | More Colors... |
| WebGoBack | button | Back |
| WebGoForward | button | Forward |
| AddToFavorites | button | Add to Favorites... |
| BrowsePrevious | button | Previous |
| BrowseNext | button | Next |
| SmartArtInsert | button | SmartArt... |
| ShapeRerouteConnectors | toggleButton | Reroute Connectors |
| ObjectNudgeUp | button | Up |
| ObjectNudgeDown | button | Down |
| ObjectNudgeLeft | button | Left |
| ObjectNudgeRight | button | Right |
| ShapeCurve | toggleButton | Curve |
| ShapeStraightConnector | toggleButton | Straight Connector |
| ShapeElbowConnector | toggleButton | Elbow Connector |
| ObjectFillMoreColorsDialog | button | More Fill Colors... |
| ObjectBorderOutlineColorMoreColorsDialog | button | More Outline Colors... |
| OutlineLinePatternFill | button | Pattern... |
| LineStylesDialog | button | More Lines... |
| ArrowsMore | button | More Arrows... |
| WordArtVerticalText | toggleButton | Vertical Text |

| idMso | Control Type | Label |
|-----------------------------------|--------------|-----------------------------|
| WordArtEvenTextHeightClassic | toggleButton | Even Height |
| ContrastMore | button | More Contrast |
| ContrastLess | button | Less Contrast |
| BrightnessMore | button | More Brightness |
| BrightnessLess | button | Less Brightness |
| ShadowNudgeUpClassic | button | Nudge Shadow Up |
| ShadowNudgeDownClassic | button | Nudge Shadow Down |
| ShadowNudgeLeftClassic | button | Nudge Shadow Left |
| ShadowNudgeRightClassic | button | Nudge Shadow Right |
| ObjectShadowColorMoreColorsDialog | button | More Shadow Colors... |
| _3DEffectColorPickerMoreClassic | button | More 3-D Colors... |
| TextAlignLeft | toggleButton | Left Align |
| TextAlignCenter | toggleButton | Center |
| ShapeRectangle | toggleButton | Rectangle |
| ShapeRoundedRectangle | toggleButton | Rounded Rectangle |
| ShapeIsoscelesTriangle | toggleButton | Isosceles Triangle |
| ShapeOval | toggleButton | Oval |
| ShapeLeftBrace | toggleButton | Left Brace |
| ShapeRightBrace | toggleButton | Right Brace |
| ShapeArc | toggleButton | Arc |
| ShapeRightArrow | toggleButton | Right Arrow |
| ShapeDownArrow | toggleButton | Down Arrow |
| ShapeRoundedRectangularCallout | toggleButton | Rounded Rectangular Callout |
| ShapeStar | toggleButton | 5-Point Star |
| TextAlignRight | toggleButton | Right Align |
| TextAlignLetterJustify | toggleButton | Letter Justify |
| TextAlignWordJustify | toggleButton | Word Justify |
| TextAlignStretchJustify | toggleButton | Stretch Justify |
| WordArtSpacingVeryTight | toggleButton | Very Tight |

| idMso | Control Type | Label |
|----------------------------------|---------------------|----------------------|
| WordArtSpacingTight | toggleButton | Tight |
| WordArtSpacingNormal | toggleButton | Normal |
| WordArtSpacingLoose | toggleButton | Loose |
| WordArtSpacingVeryLoose | toggleButton | Very Loose |
| WordArtSpacingKernCharacterPairs | toggleButton | Kern Character Pairs |
| PictureReset | button | Reset Picture |
| TextWrappingSquare | toggleButton | Square |
| TextWrappingTight | toggleButton | Tight |
| TextWrappingNoneClassic | toggleButton | None |
| TextWrappingEditWrapPoints | toggleButton | Edit Wrap Points |
| _3DEffectsOnOffClassic | toggleButton | 3-D On/Off |
| _3DTiltDownClassic | button | Tilt Down |
| _3DTiltUpClassic | button | Tilt Up |
| _3DTiltLeftClassic | button | Tilt Left |
| _3DTiltRightClassic | button | Tilt Right |
| _3DExtrusionPerspectiveClassic | toggleButton | Perspective |
| _3DExtrusionParallelClassic | toggleButton | Parallel |
| _3DLightingFlatClassic | toggleButton | Bright |
| _3DLightingNormalClassic | toggleButton | Normal |
| _3DLightingDimClassic | toggleButton | Dim |
| ObjectEditText | button | Edit Text |
| PictureFormatDialog | button | Picture... |
| ViewVisualBasicCode | button | View Code |
| DrawingNewClassic | button | New Drawing |
| WebOpenInNewWindow | button | Open in New Window |
| HyperlinkCopy | button | Copy Hyperlink |
| HyperlinkInsert | button | Hyperlink... |
| HyperlinkEdit | button | Edit Hyperlink... |
| HyperlinkSelect | button | Select Hyperlink |

| idMso | Control Type | Label |
|--|---------------------|------------------------------|
| ReviewNewComment | button | New Comment |
| ReviewPreviousComment | button | Previous |
| ReviewNextComment | button | Next |
| ReviewDeleteComment | button | Delete |
| ReviewShowAllComments | button | Show All Comments |
| DesignMode | toggleButton | Design Mode |
| WordArtInsertDialogClassic | button | WordArt Gallery |
| FormFieldProperties | button | Properties |
| FullScreenViewClassic | button | Full Screen |
| AutoScroll | button | Auto Scroll |
| MasterDocumentExpandOrCollapseSubdocuments | toggleButton | Expand/Collapse Subdocuments |
| VisualBasic | button | Visual Basic |
| BordersAll | toggleButton | All Borders |
| AutoSummarize | button | Auto Summarize |
| ViewDocumentMap | checkBox | Document Map |
| ReviewAcceptChange | button | Accept Change |
| ReviewRejectChange | button | Reject Change |
| TableDrawBorderPenStyle | dropDown | Pen Style |
| AutoSummaryExitView | button | Close |
| Font | comboBox | Font: |
| WhoIs | button | Who Is... |
| FontSize | comboBox | Font Size: |
| StyleGalleryClassic | comboBox | Style: |
| ZoomClassic | button | Zoom: |
| DocumentLocation | comboBox | Address: |
| MessageHeaderToggle | button | Message Header |
| BorderInsideHorizontal | toggleButton | Inside Horizontal Border |
| BorderInsideVertical | toggleButton | Inside Vertical Border |
| BorderDiagonalDown | toggleButton | Diagonal Down Border |

| idMso | Control Type | Label |
|--------------------------------|---------------------|------------------------|
| BorderDiagonalUp | toggleButton | Diagonal Up Border |
| TextDirectionLeftToRight | toggleButton | Left-to-Right |
| TextDirectionRightToLeft | toggleButton | Right-to-Left |
| ActiveXCheckBox | button | Check Box |
| ActiveXTextBox | button | Text Box |
| ActiveXButton | button | Command Button |
| ActiveXRadioButton | button | Option Button |
| ActiveXListBox | button | List Box |
| ActiveXComboBox | button | Combo Box |
| ActiveXToggleButton | button | Toggle Button |
| ActiveXSpinButton | button | Spin Button |
| ActiveXScrollBar | button | Scroll Bar |
| ActiveXLabel | button | Label |
| ShadowSemitransparentClassic | toggleButton | Semitransparent Shadow |
| OleConvert | button | Convert... |
| ReviewTrackChanges | toggleButton | Track Changes |
| ReviewHighlightChanges | button | Highlight Changes... |
| ReviewEditComment | button | Edit Comment |
| TableDrawTable | toggleButton | Draw Table |
| TableEraser | toggleButton | Eraser |
| TableCellAlignTop | toggleButton | Align Top |
| TableCellAlignCenterVertically | toggleButton | Center Vertically |
| TableCellAlignBottom | toggleButton | Align Bottom |
| TableColumnsDistribute | button | Distribute Columns |
| TableRowsDistribute | button | Distribute Rows |
| ActiveXFrame | button | Frame |
| ActiveXImage | button | Image |
| WordArtEditTextClassic | button | Edit Text... |
| TableInsertCellsDialog | button | Insert Cells... |

| idMso | Control Type | Label |
|----------------------------|---------------------|-----------------------------|
| Organizer | button | Organizer |
| ShadowOnOrOffClassic | toggleButton | Shadow On/Off |
| ObjectSetShapeDefaults | button | Set AutoShape Defaults |
| ThesaurusClassic | button | Thesaurus... |
| MacroRecorderStop | button | Stop Recording |
| FileSendAsAttachment | button | E-mail |
| AutoSummaryViewByHighlight | toggleButton | Highlight/Show Only Summary |
| MasterDocument | button | Master Document |
| SystemInformation | button | Microsoft System Info |
| Overtyp | button | Overtyp |
| ExtendSelection | button | Extend Selection |
| Spike | button | Spike |
| SpikeInsert | button | Insert Spike |
| ChangeCase | button | Change Case |
| MoveText | button | Move Text |
| CopyText | button | Copy Text |
| AutoTextInsert | button | Insert AutoText |
| WindowOtherPane | button | Other Pane |
| WindowPrevious | button | Previous Window |
| FieldNext | button | Next Field |
| FieldPrevious | button | Previous Field |
| TableColumnSelectWord | button | Column Select |
| FieldCharactersInsert | button | Insert Field Chars |
| ListNumFieldInsert | button | Insert ListNum Field |
| FieldsUnlink | button | Unlink Fields |
| FieldsLock | button | Lock Fields |
| FieldsUnlock | button | Unlock Fields |
| UpdateSource | button | Update Source |
| HangingIndent | button | Hanging Indent |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|-------------------------------|
| UnHang | button | Un Hang |
| HideText | button | Hidden |
| FontSpacingNormal | button | Normal Font Spacing |
| FontPositionNormal | button | Normal Font Position |
| ParagraphWidowOrphanControl | button | Para Widow Orphan Control |
| ParagraphKeepLinesTogether | button | Para Keep Lines Together |
| ParagraphKeepWithNext | toggleButton | Para Keep With Next |
| BreakParagraphPageBreakBefore | button | Para Page Break Before |
| ParagraphSpaceBeforeNone | button | No Space Before |
| ParagraphSpaceBefore | button | Space Before |
| ParagraphSpaceAddOrRemoveBefore | button | Add/Remove Space Before |
| ParagraphReset | button | Reset Para |
| PreviousEdit | button | Previous Edit |
| NextEdit | button | Next Edit |
| SaveTemplate | button | Save Template |
| PagePreviousWord | button | Previous Page |
| PageNextWord | button | Next Page |
| ObjectNext | button | Next Object |
| ObjectPrevious | button | Previous Object |
| FileConfirmConversions | button | File Confirm Conversions |
| MailMergeReceipientsUseExistingList | button | Use Existing List... |
| MailMergeOpenHeaderSource | button | Mail Merge Open Header Source |
| MailMergeQueryOptions | button | Query Options |
| MailMergeRuleIfThenElse | button | Mail Merge Insert If |
| MailMergeRuleMergeRecordNumber | button | Mail Merge Insert Merge Rec |
| MailMergeRuleMergeSequenceNumber | button | Mail Merge Insert Merge Seq |
| MailMergeRuleNextRecord | button | Mail Merge Insert Next |

| idMso | Control Type | Label |
|-----------------------------|---------------------|---------------------------------|
| MailMergeRuleNextRecordIf | button | Mail Merge Insert Next If |
| MailMergeRuleSkipRecordIf | button | Mail Merge Insert Skip If |
| MailMergeRuleFillIn | button | Mail Merge Insert Fill In |
| MailMergeRuleAsk | button | Mail Merge Insert Ask |
| MailMergeRuleSetBookmark | button | Mail Merge Insert Set |
| MailMergeReset | button | Mail Merge Reset |
| MailMergeCreateDataSource | button | Mail Merge Create Data Source |
| MailMergeCreateHeaderSource | button | Mail Merge Create Header Source |
| GoToPreviousSection | button | Go To Previous Section |
| GoToNextSection | button | Go To Next Section |
| GoToPreviousPage | button | Go To Previous Page |
| GoToNextPage | button | Go To Next Page |
| FootnotePreviousWord | button | Previous Footnote |
| FootnoteNextWord | button | Next Footnote |
| EndnotePreviousWord | button | Previous Endnote |
| EndnoteNextWord | button | Next Endnote |
| ObjectActivate | button | Activate Object |
| TableAutoFormatUpdate | button | Table Update AutoFormat |
| DraftViewClassic | button | View Draft |
| NormalViewHeaderArea | button | Normal View Header Area |
| SectionBreakInsert | button | Insert Section Break |
| EndnoteInsertWord | button | Insert Endnote |
| FootnotesConvertAll | button | Edit Convert All Footnotes |
| EndnotesConvertAll | button | Edit Convert All Endnotes |
| SwapAllNotes | button | Edit Swap All Notes |
| InsertEnSpace | button | Insert En Space |
| InsertEmSpace | button | Insert Em Space |
| IndexMarkEntry | button | Mark Entry... |

| idMso | Control Type | Label |
|--------------------------------|---------------------|--------------------------------|
| AutoMarkIndexEntries | button | Auto Mark Index Entries |
| CitationMark | button | Mark Citation... |
| TableOfAuthoritiesEditCategory | button | Edit TOA Category |
| IndexInsert | button | Insert Index... |
| TableOfContentsDialog | button | Insert Table of Contents... |
| TableOfContentsMarkEntry | button | Mark Entry |
| TableOfFiguresInsert | button | Insert Table of Figures... |
| TableOfAuthoritiesInsert | button | Insert Table of Authorities... |
| DrawingUnselect | button | Draw Unselect |
| DrawingSelectNext | button | Draw Select Next |
| DrawingSelectPrevious | button | Draw Select Previous |
| TextBoxLinkCreate | button | Create Link |
| TextBoxLinkBreak | button | Break Link |
| TextBoxNextLinked | button | Next Text Box |
| TextBoxPreviousLinked | button | Previous Text Box |
| FormatSectionLayout | button | Format Section Layout |
| StylesRedefineStyle | button | Redefine Style |
| Heading1Apply | button | Apply Heading 1 |
| Heading2Apply | button | Apply Heading 2 |
| Heading3Apply | button | Apply Heading 3 |
| ListBulletApply | button | Apply List Bullet |
| TextBoxConvertToFrame | button | Convert Text Box To Frame |
| ListPromote | button | Promote List |
| ListDemote | button | Demote List |
| NextMisspeling | button | Next Misspelling |
| HyphenationManual | button | Manual |
| BulletsAndNumberingClassic | button | Tools Bullets Numbers |
| CompareAndCombine | button | Compare & Combine |

| idMso | Control Type | Label |
|-------------------------------|---------------------|--------------------------------------|
| Calculate | button | Tools Calculate |
| KeyboardCustomizationWord | button | Tools Customize Keyboard Shortcut |
| ListCommands | button | List Commands |
| PrintOptionsMenuWord | button | Options |
| SpellingRecheckDocument | button | Tools Spelling Recheck Document |
| ReviewChangeUserName | button | Change User Name... |
| AutoFormatOptions | button | Tools Options AutoFormat |
| AutoFormatAsYouType | button | Tools Options AutoFormat As You Type |
| MailMergeConvertChevrons | button | Mail Merge Convert Chevrons |
| MailMergeAskToConvertChevrons | button | Mail Merge Ask To Convert Chevrons |
| ControlRun | button | Control Run |
| ShrinkSelection | button | Shrink Selection |
| StyleNormal | button | Normal Style |
| TableCellNext | button | Next Cell |
| TableCellPrevious | button | Previous Cell |
| StartOfRow | button | Start Of Row |
| EndOfRow | button | End Of Row |
| StartOfColumn | button | Start Of Column |
| EndOfColumn | button | End Of Column |
| WindowMinimizeAll | button | Minimize All |
| WindowMaximizeAll | button | Maximize All |
| WindowRestoreAll | button | Restore All |
| FieldClick | button | Do Field Click |
| SelectCurrentFont | button | Select Cur Font |
| SelectCurrentAlignment | button | Select Cur Alignment |
| SelectCurrentSpacing | button | Select Cur Spacing |
| SelectCurrentIndent | button | Select Cur Indent |

| idMso | Control Type | Label |
|-------------------------------|---------------------|---------------------------------|
| SelectCurrentTabs | button | Select Cur Tabs |
| SelectCurrentColor | button | Select Cur Color |
| FramesRemove | button | Remove Frames |
| MenuMode | button | Menu Mode |
| PageNumberFormat | button | Format Page Numbers... |
| Zoom200 | button | View Zoom200 |
| Zoom75 | button | View Zoom75 |
| AddressFontsFormat | button | Format Addr Fonts |
| ReturnAddressFormatFontDialog | button | Format Ret Addr Fonts |
| FileLocations | button | Tools Options File Locations |
| CreateDirectoryClassic | button | Tools Create Directory |
| TableOfContentsUpdateClassic | button | Update Table of Contents... |
| FootnoteSeparatorWord | button | View Footnote Separator |
| FootnoteContinuationSeparator | button | View Footnote Cont Separator |
| FootnoteContinuationNotice | button | View Footnote Cont Notice |
| EndnoteSeparator | button | View Endnote Separator |
| EndnoteContinuationSeparator | button | View Endnote Cont Separator |
| EndnoteContinuationNotice | button | View Endnote Cont Notice |
| AutoCaptionInsert | button | Insert Auto Caption |
| CaptionInsertWord | button | Insert Add Caption |
| InsertCaptionNumbering | button | Insert Caption Numbering |
| AutoCorrectReplaceText | button | Tools AutoCorrect Replace Text |
| AutoCorrectInitialCaps | button | Tools AutoCorrect Initial Caps |
| AutoCorrectSentenceCaps | button | Tools AutoCorrect Sentence Caps |
| AutoCorrectDays | button | Tools AutoCorrect Days |
| AutoCorrectSmartQuotes | button | Tools AutoCorrect Smart |

| idMso | Control Type | Label |
|------------------------------|--------------|---------------------------------|
| | | Quotes |
| AutoCorrectCapsLockOff | button | Tools AutoCorrect Caps Lock Off |
| AutoCorrectExceptions | button | Tools AutoCorrect Exceptions |
| WindowSizeAll | button | Size All |
| WindowMoveAll | button | Move All |
| ConnectToNetworkDrive | button | Connect |
| GoToAnnotationScope | button | Goto Annotation Scope |
| FontSubstitution | button | Font Substitution |
| ScreenRefresh | button | Screen Refresh |
| CharacterLeft | button | Char Left |
| CharacterRight | button | Char Right |
| WordLeft | button | Word Left |
| WordRight | button | Word Right |
| ExtendSelectionLeft | button | Sent Left |
| ExtendSelectionRight | button | Sent Right |
| ParagraphUp | button | Para Up |
| ParagraphDown | button | Para Down |
| LineUp | button | Line Up |
| LineDown | button | Line Down |
| CharacterLeftExtend | button | Char Left Extend |
| CharacterRightExtend | button | Char Right Extend |
| WordLeftExtend | button | Word Left Extend |
| WordRightExtend | button | Word Right Extend |
| ExtendSelectionLeftSentence | button | Sent Left Extend |
| ExtendSelectionRightSentence | button | Sent Right Extend |
| ParagraphUpExtend | button | Para Up Extend |
| ParagraphDownExtend | button | Para Down Extend |
| LineUpExtend | button | Line Up Extend |
| LineDownExtend | button | Line Down Extend |

| idMso | Control Type | Label |
|-----------------------------|---------------------|------------------------------|
| PageUpExtend | button | Page Up Extend |
| PageDownExtend | button | Page Down Extend |
| StartOfLineExtend | button | Start Of Line Extend |
| EndOfLineExtend | button | End Of Line Extend |
| StartOfWindowExtend | button | Start Of Window Extend |
| EndOfWindowExtend | button | End Of Window Extend |
| StartOfDocumentExtend | button | Start Of Doc Extend |
| EndOfDocumentExtend | button | End Of Doc Extend |
| SymbolFont | button | Symbol Font |
| GrammarSettingsDialog | button | Tools Gram Settings |
| FileNewDefault | button | New |
| FilePrintQuick | button | Quick Print |
| SpellingAndGrammar | button | Spelling & Grammar |
| ReviewPreviousChangeClassic | button | Previous Change |
| ReviewNextChangeClassic | button | Next Change |
| MessageProperties | button | Properties |
| PictureInsertFromFile | button | Picture... |
| TableDrawBorderPenWeight | dropDown | Pen Weight |
| TableShowGridlines | toggleButton | View Gridlines |
| ShapeStraightConnectorArrow | toggleButton | Straight Arrow Connector |
| ShapeElbowConnectorArrow | toggleButton | Elbow Arrow Connector |
| HyperlinkOpen | button | Open Hyperlink |
| TextWrappingTopAndBottom | toggleButton | Top and Bottom |
| TextWrappingThrough | toggleButton | Through |
| MacroRecordOrStop | button | Record Macro / Stop Recorder |
| AutoManager | button | AutoManager... |
| EndnoteOrFootnoteConvert | button | Convert Endnote/Footnote |
| FootnoteSeparatorReset | button | Reset |
| PasteAsHyperlink | button | Paste as Hyperlink |

| idMso | Control Type | Label |
|----------------------------------|---------------------|---------------------------------|
| ProofingOptions | button | Options... |
| ParagraphDistributed | toggleButton | Distributed |
| HyphenationOptions | button | Hyphenation Options... |
| TableRowsOrColumnsDistribute | button | Distribute Rows/Columns |
| MergeOrSplitCells | button | Merge/Split Cells |
| ReviewJapaneseConsistencyChecker | button | Japanese Consistency Checker... |
| AutoSummaryResummarize | button | Resummarize |
| AutoSummaryUpdateProperties | button | Update Properties |
| DeleteWord | button | Delete Word |
| DeleteWordBack | button | Delete Back Word |
| CharacterFormattingReset | button | Reset Character Formatting |
| HeadingNumbers | button | Heading Numbers |
| PictureSetTransparentColor | toggleButton | Set Transparent Color |
| PageColorMoreColorsDialog | button | More Colors... |
| PageColorFillEffects | button | Fill Effects... |
| BorderTopWord | toggleButton | Top Border |
| BorderBottomWord | toggleButton | Bottom Border |
| BorderLeftWord | toggleButton | Left Border |
| BorderRightWord | toggleButton | Right Border |
| TextDirection | button | Text Direction |
| FieldsManage | button | Manage |
| FileSaveAsHtml | button | Save as HTML... |
| SortAscendingWord | button | Sort Ascending |
| SortDescendingWord | button | Sort Descending |
| FrameInsertHorizontal | button | Horizontal Frame |
| FieldCodesToggle | button | Toggle Field Codes |
| GoToFootnote | button | Go to Footnote |
| GoToEndnote | button | Go to Endnote |

| idMso | Control Type | Label |
|---------------------------|---------------------|-------------------------------------|
| SpellingHideErrors | toggleButton | Hide Spelling Errors |
| GrammarHideErrors | toggleButton | Hide Grammar Errors |
| Dictionary | button | Dictionary |
| SummaryInformation | button | Summary Information... |
| FootnoteEndnoteOptions | button | Footnote/Endnote Options |
| UnderlineDotted | toggleButton | Dotted Underline |
| NumberingRemove | button | Remove Numbering |
| PictureEditClassic | button | Picture |
| GoToStartOfWindow | button | Start of Window |
| GoToEndOfWindow | button | End of Window |
| AutoCorrectHECorrect | button | HECorrect |
| FileSendToPowerPoint | button | Send to Microsoft Office PowerPoint |
| FormatObjectDialogClassic | button | AutoShape... |
| AutoFormatNow | button | AutoFormat... |
| DataFormWord | button | Data Form |
| BulletListDefault | button | Tools Bullet List Default |
| NumberListDefault | button | Tools Number List Default |
| OutlineNumberDefault | button | Format Outline Number Default |
| FormatNumberDefault | button | Format Number Default |
| TableOfContentsRebuild | button | Rebuild Table of Contents |
| FootnoteEndnoteDialog | button | Footnote and Endnote Dialog... |
| TableInsertDialogWord | button | Insert Table... |
| FormFieldClear | button | Clear Form Field |
| ObjectBringInFrontOfText | button | Bring in Front of Text |
| ObjectSendBehindText | button | Send Behind Text |
| PageBreakInsertWord | button | Page Break |
| BordersShadingDialogWord | button | Borders and Shading... |
| TextBoxWordClassic | button | Text Box |

| idMso | Control Type | Label |
|------------------------------|---------------------|----------------------------|
| IndentIncreaseWord | button | Increase Indent |
| IndentDecreaseWord | button | Decrease Indent |
| SelectObjects | button | Select Objects |
| Callout | button | Callout |
| ReplaceWithAutoText | button | Replace with AutoText |
| LinkToPreviousClassic | button | Link to Previous |
| HangulHanjaConversion | button | Hangul Hanja Conversion... |
| HeaderSourceEdit | button | Edit Header Source |
| IndentIncrease | button | Increase Indent |
| IndentDecrease | button | Decrease Indent |
| AsianLayoutFitText | button | Fit Text... |
| AsianLayoutPhoneticGuide | button | Phonetic Guide... |
| AsianLayoutCombineCharacters | button | Combine Characters... |
| JapanesePostcardDialog | button | Japanese Postcard... |
| CharacterBorder | toggleButton | Character Border |
| CharacterShading | toggleButton | Character Shading |
| ViewWebLayoutView | toggleButton | Web Layout |
| PasteAlternative | button | Paste Table |
| PasteAsNestedTable | button | Paste as Nested Table |
| HyperlinkRemove | button | Remove Hyperlink |
| MacroSecurity | button | Macro Security |
| HorizontalLineInsert | button | Horizontal Line |
| WebPagePreview | button | Web Page Preview |
| RightToLeftRun | button | Rtl Run |
| LeftToRightRun | button | Ltr Run |
| BoldRun | button | Bold Run |
| ItalicRun | button | Italic Run |
| TableSelectCell | button | Select Cell |
| TableDelete | button | Delete Table |

| idMso | Control Type | Label |
|-----------------------------|--------------|----------------------------|
| TableRowsInsertAboveWord | button | Insert Above |
| TableRowsInsertBelowWord | button | Insert Below |
| TableColumnsInsertLeft | button | Insert Left |
| TableColumnsInsertRight | button | Insert Right |
| TablePropertiesDialog | button | Properties... |
| TableOptionsDialog | button | Cell Margins... |
| TableCellOptions | button | Cell Options... |
| SendCopySendNow | button | Send Now |
| SendCopySelectNames | button | Select Names |
| SendCopyCheckNames | button | Check Names |
| SendCopyAddressBookTo | button | To: Focus |
| SendCopyAddressBookCc | button | CC: Focus |
| SendCopyAddressBookBcc | button | Bcc: Focus |
| SendCopyFocusSubject | button | Subject Focus |
| SendCopyOptions | button | Mail Options |
| SendCopyFlag | button | Mail Flag |
| SendCopySaveAttachment | button | Save Mail Attachments |
| FileCloseOrExit | button | Close or Exit |
| ImeReconvert | button | Reconvert |
| SendCopySendToMailRecipient | toggleButton | Mail Recipient |
| TableOfContentsInFrame | button | Table of Contents in Frame |
| SetLanguageMenu | comboBox | Language |
| TableWrapping | button | Table Wrapping |
| EmailOptions | button | E-mail Options... |
| ComAddInsDialog | button | COM Add-Ins... |
| SignaturesStationeryDialog | button | Signatures... |
| FramePropertiesDialog | button | Frame Properties... |
| OfficeOnTheWeb | button | Microsoft Office Online |
| PictureBulletsInsert | button | Picture Bullets... |

| idMso | Control Type | Label |
|------------------------------|---------------------|-----------------------------|
| FileNewWebPage | button | New Web Page |
| FileNewBlankDocument | button | New Blank Document |
| FileNewDialogClassic | button | New Document or Template... |
| FileSaveAsWebPage | button | Save as Web Page... |
| HorizontalLineInsertClassic | button | Horizontal Line... |
| WebOptionsDialog | button | Web Options... |
| FramesNewFramesPageWizard | button | New Frames Page |
| FrameCreateAbove | button | New Frame Above |
| FrameCreateBelow | button | New Frame Below |
| FrameCreateLeft | button | New Frame Left |
| FrameCreateRight | button | New Frame Right |
| FrameDelete | button | Delete Frame |
| EastAsianEditingMarks | toggleButton | Show/Hide Editing Marks |
| TableAutoFitContents | button | AutoFit Contents |
| TableAutoFitWindow | button | AutoFit Window |
| TableAutoFitFixedColumnWidth | button | Fixed Column Width |
| TableCellAlignTopLeft | toggleButton | Align Top Left |
| TableCellAlignTopCenter | toggleButton | Align Top Center |
| TableCellAlignTopRight | toggleButton | Align Top Right |
| TableCellAlignMiddleLeft | toggleButton | Align Center Left |
| TableCellAlignMiddleCenter | toggleButton | Align Center |
| TableCellAlignMiddleRight | toggleButton | Align Center Right |
| TableCellAlignBottomLeft | toggleButton | Align Bottom Left |
| TableCellAlignBottomCenter | toggleButton | Align Bottom Center |
| TableCellAlignBottomRight | toggleButton | Align Bottom Right |
| WebControlCheckBox | button | Checkbox |
| WebControlOptionButton | button | Option Button |
| WebControlDropDownBox | button | Drop-Down Box |
| WebControlListBox | button | List Box |

| idMso | Control Type | Label |
|---------------------------------|--------------|------------------------------|
| WebControlTextBox | button | Textbox |
| WebControlTextArea | button | Text Area |
| WebControlSubmit | button | Submit |
| WebControlSubmitWithImage | button | Submit with Image |
| WebControlReset | button | Reset |
| WebControlHidden | button | Hidden |
| WebControlPassword | button | Password |
| UnderlineColorMoreColorsDialog | button | More Colors... |
| ChineseTranslationDialog | button | Translate with Options... |
| TableInsertMultidiagonalCell | button | Insert Multidiagonal Cell... |
| AsianLayoutHorizontalInVertical | button | Horizontal in Vertical... |
| AsianLayoutTwoLinesInOne | button | Two Lines in One... |
| AsianLayoutCharactersEnclose | button | Enclose Characters... |
| EnvelopeChineseDialog | button | Chinese Envelope... |
| ObjectsMultiSelect | button | Select Multiple Objects |
| TranslateToTraditionalChinese | button | Traditional |
| TextWrappingBehindText | toggleButton | Behind Text |
| TextWrappingInFrontOfText | toggleButton | In Front of Text |
| WatermarkCustomDialog | button | Custom Watermark... |
| FrameSaveCurrentAs | button | Save Current Frame As... |
| TranslateToSimplifiedChinese | button | Simplified |
| WebDesignMode | toggleButton | Web Design Mode |
| ViewMasterDocumentViewClassic | button | Master Document Tools |
| WhiteSpaceBetweenPagesShowHide | button | White Space Between Pages |
| EditField | button | Edit Field... |
| WordCountRecount | button | Recount |
| StylesModifyStyle | button | Modify Style |
| StyleByExample | button | Style by Example |
| CssLinksEdit | button | Edit CSS Links |

| idMso | Control Type | Label |
|------------------------------------|---------------------|--------------------------|
| StylesPane | button | Styles... |
| DeleteStyle | button | Delete Style |
| StylesRenameStyle | button | Rename Style |
| SelectNumber | button | Select Number |
| NumberingRestart | button | Restart Numbering |
| DrawingCanvasInsert | button | New Drawing Canvas |
| DiagramRadialInsertClassic | button | Radial Diagram |
| DiagramCycleInsertClassic | button | Cycle Diagram |
| DiagramPyramidInsertClassic | button | Pyramid Diagram |
| DiagramTargetInsertClassic | button | Target Diagram |
| DiagramVennDiagramInsertClassic | button | Venn Diagram |
| DiagramChangeToRadialClassic | button | Radial |
| DiagramChangeToCycleClassic | button | Cycle |
| DiagramChangeToTargetClassic | button | Target |
| DiagramChangeToVennDiagramClassic | button | Venn |
| CopyPasteSettings | button | Copy & paste settings... |
| PasteByAppendingTable | button | Paste by Appending Table |
| OrganizationChartInsertAssistant | button | Assistant |
| OrganizationChartInsertCoworker | button | Coworker |
| OrganizationChartInsertSubordinate | button | Subordinate |
| OrganizationChartDeleteNode | button | Delete |
| DrawingCanvasFit | button | Fit |
| DrawingCanvasResize | button | Resize |
| LabelOptions | button | Label Options... |
| SendCopySetup | button | Envelope Setup... |
| MailMergeMergeToEMail | button | Send E-mail Messages... |
| MailMergeMergeToFax | button | Merge to Fax |
| MailMergeCreateList | button | Type New List... |
| MailMergeEditList | button | Edit Mail Merge List |

| idMso | Control Type | Label |
|--|---------------------|-------------------------------------|
| DrawingCanvasExpand | button | Expand |
| ActivateProduct | button | Activate Product... |
| TextWrappingInLineWithText | toggleButton | In Line with Text |
| ConsistencyCheck | button | Consistency Check... |
| SelectTextWithSimilarFormatting | button | Select Text with Similar Formatting |
| ReviewSendForReview | button | Send for Review... |
| WebComponent | button | Web Component... |
| DiagramChangeToPyramidClassic | button | Pyramid |
| DiagramShapeMoveBackwardClassic | button | Move Shape Backward |
| DiagramShapeMoveForwardClassic | button | Move Shape Forward |
| CharacterCodeToggle | button | Toggle Character Code |
| SmartTagOptions | button | Tools Options Smart Tag |
| SendCopyFocusIntroduction | button | Introduction Focus |
| StylesStyleVisibility | button | Style Visibility |
| DiagramStylesClassic | button | Diagram Styles... |
| MailMergeHighlightMergeFields | toggleButton | Highlight Merge Fields |
| MailMergeWizard | toggleButton | Step by Step Mail Merge Wizard... |
| OrganizationChartAutoLayout | toggleButton | AutoLayout |
| OrganizationChartSelectLevel | button | Level |
| OrganizationChartSelectBranch | button | Branch |
| OrganizationChartSelectAllAssistants | button | All Assistants |
| OrganizationChartSelectAllConnectors | button | All Connectors |
| MailMergeJapaneseGreetingInsert | button | Greeting... |
| MailMergeJapaneseGreetingJapaneseOpeningSentenceInsert | button | Opening... |
| MailMergeJapaneseGreetingClosingSentenceInsert | button | Closing... |
| RevealFormatting | button | Reveal Formatting... |
| DiagramReverseClassic | button | Reverse |
| DiagramAutoLayoutClassic | toggleButton | AutoLayout |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|--------------------------|
| TextBoxAutosize | button | Autosize textbox |
| TranslationPane | button | Translate... |
| GoToTableOfContents | button | Go to TOC |
| TableOfContentsUpdate | button | Update Table... |
| OutlineLevelGallery | dropDown | Outline Level |
| OutlineShowLevel | dropDown | Show Level: |
| NumberingContinue | button | Continue Numbering |
| FileCheckOut | button | Check Out |
| FileCheckIn | button | Check In |
| OrganizationChartLayoutStandard | toggleButton | Standard |
| OrganizationChartLayoutBothHanging | toggleButton | Both Hanging |
| OrganizationChartLayoutLeftHanging | toggleButton | Left Hanging |
| OrganizationChartLayoutRightHanging | toggleButton | Right Hanging |
| ReviewShowReviewersMenu | menu | Reviewers |
| ReviewReplyWithChanges | button | Reply with Changes... |
| ReviewEndReview | button | End Review... |
| NormalizeText | button | Normalize text |
| StylesStyleSeparator | button | Style Separator |
| SpeakLearnFromDocument | button | Learn from Document... |
| PictureEditWord | button | Edit Picture |
| OutlinePromoteToHeading | button | Promote to Heading 1 |
| MicrosoftOutlook | button | Microsoft Outlook |
| ReviewShowComments | toggleButton | Comments |
| ReviewShowInsertionsAndDeletions | toggleButton | Insertions and Deletions |
| ReviewShowFormatting | toggleButton | Formatting |
| ReviewPreviousChange | button | Previous |
| ReviewNextChange | button | Next |
| ReviewReviewingPane | toggleButton | Reviewing Pane |
| ReviewAcceptAllChangesShown | button | Accept All Changes Shown |

| idMso | Control Type | Label |
|------------------------------------|---------------------|------------------------------------|
| ReviewAcceptAllChangesInDocument | button | Accept All Changes in Document |
| ReviewRejectAllChangesShown | button | Reject All Changes Shown |
| ReviewRejectAllChangesInDocument | button | Reject All Changes in Document |
| ReviewDeleteAllCommentsShown | button | Delete All Comments Shown |
| ReviewDeleteAllCommentsInDocument | button | Delete All Comments in Document |
| ShowRepairs | button | Show Repairs |
| MailMergeMatchFields | button | Match Fields... |
| MailMergeAddressBlockInsert | button | Address Block... |
| MailMergeGreetingLineInsert | button | Greeting Line... |
| MailMergeMergeFieldInsert | button | Insert Merge Field |
| MailMergeRecipientsEditList | button | Edit Recipient List... |
| MailMergeEmailOptions | button | Mail Merge E-Mail Options |
| MailMergePrintOptions | button | Mail Merge Print Options |
| MailMergeFaxOptions | button | Mail Merge Fax Options |
| MailMergeMergeToNewDocumentOptions | button | Mail Merge to New Document Options |
| PicturesCompress | button | Compress Pictures... |
| Security | button | Security |
| TableAutoFormatStyle | button | Table AutoFormat... |
| MailMergeEditAddressBlock | button | Edit Address Block... |
| MailMergeEditGreetingLine | button | Edit Greeting Line... |
| MailMergeFindRecipient | button | Find Recipient... |
| FormFieldReset | button | Reset Form Fields |
| MailMergeUpdateLabels | button | Update Labels |
| DiagramFitToContentsClassic | button | Fit to Contents |
| DiagramResizeClassic | toggleButton | Resize |
| DiagramExpandClassic | button | Expand |
| OrganizationChartResize | button | Resize Organization Chart |

| idMso | Control Type | Label |
|--------------------------------------|---------------------|---------------------------|
| AccountSettings | button | Account Settings... |
| MailMergeSetDocumentType | button | Main document setup |
| DiagramShapeInsertClassic | button | Insert Shape |
| OrganizationChartStyle | button | Style... |
| ReviewDisplayForReview | dropDown | Display for Review |
| DiagramAutoFormatClassic | button | Use AutoFormat |
| Translate | button | Translate... |
| ClearContentsWord | button | Contents |
| DrawingCanvasScale | button | Scale Drawing |
| ProtectDocument | toggleButton | Protect Document... |
| XmlViewStructure | button | View XML Structure |
| ReadingViewClose | button | Edit |
| ResearchPane | toggleButton | Research... |
| DocumentMapReadingView | button | Document Map |
| ReadingViewResearchPane | button | Research... |
| ReadingViewFontSizeIncrease | button | Increase Text Size |
| ReadingViewFontSizeDecrease | button | Decrease Text Size |
| ReadingViewShowPrintedPage | button | Show Printed Page |
| ViewRulerWord | checkBox | Ruler |
| FileInternetFax | button | Internet Fax |
| DocumentUpdatePane | button | Show Document Update Pane |
| ViewDocumentActionsPane | toggleButton | Document Actions |
| XmlToggleTagView | button | Toggle XML Tag View |
| InkDeleteAllInk | button | Delete All Ink |
| LookUp | button | Look Up |
| WindowSideBySide | toggleButton | View Side by Side |
| PrivacyOptionsDialog | button | Privacy Options... |
| FileVersionHistoryWord | button | View Version History |
| WindowSideBySideSynchronousScrolling | toggleButton | Synchronous Scrolling |

| idMso | Control Type | Label |
|---|--------------|---|
| WindowResetPosition | button | Reset Window Position |
| InkColorMoreColorsDialog | button | More Ink Colors... |
| XmlOptionsDialog | button | XML Options... |
| XmlTransformation | toggleButton | Transformation |
| StyleEnforcementSettings | button | Style Enforcement Settings |
| ContactUs | button | Contact Us... |
| FilePermissionUnrestricted | toggleButton | Unrestricted Access |
| FilePermissionDoNotDistribute | toggleButton | Restricted Access |
| FilePermissionView | button | View Permission |
| FilePermission | button | Permission |
| ReadingViewAllowMultiplePages | button | Allow Multiple Pages |
| ReadingViewStartInking | button | Ink |
| ReadingViewUnlockDocumentLayout | button | Unlock Document Layout |
| VoiceInsertInComment | button | Insert Voice |
| ViewThumbnails | checkBox | Thumbnails |
| Thesaurus | button | Thesaurus... |
| InkingStart | button | Start Inking |
| ReviewShowRevisionsInBalloons | toggleButton | Show Revisions in Balloons |
| ReviewShowRevisionsInline | toggleButton | Show All Revisions Inline |
| ReviewShowOnlyCommentsAndFormattingInBalloons | toggleButton | Show Only Comments and Formatting in Balloons |
| ReviewShowInkMarkup | toggleButton | Ink |
| CheckForUpdates | button | Check for Updates |
| ViewFullScreenReadingView | toggleButton | Full Screen Reading |
| InkCopyAsText | button | Copy Ink As Text |
| InkDrawingAndWriting | button | Ink Drawing and Writing |
| ReviewInkCommentPen | toggleButton | Pen |
| ReviewInkCommentEraser | toggleButton | Eraser |
| InkEraser | toggleButton | Eraser |
| SendCopyAttachmentOptions | button | Attachment Options |

| idMso | Control Type | Label |
|-------------------------------|--------------|------------------------|
| FilePermissionRestrictAs | button | Manage Credentials |
| ListSetNumberingValue | button | Set Numbering Value... |
| FileViewDigitalSignatures | toggleButton | View Signatures |
| FileWorkflowTasks | button | View Workflow Tasks |
| FileStartWorkflow | button | Workflows |
| SignatureLineInsert | button | Signature Line |
| BibliographyInsert | button | Insert Bibliography |
| BibliographyStyle | comboBox | Style: |
| CitationInsert | gallery | Insert Citation |
| BibliographyManageSources | button | Manage Sources... |
| BibliographyAddNewSource | button | Add New Source... |
| LabelInsert | button | Label |
| BarcodeInsert | button | Barcode |
| ReviewShowMarkupAreaHighlight | toggleButton | Markup Area Highlight |
| ChartStylesGallery | gallery | Quick Styles |
| ChartLayoutGallery | gallery | Quick Layout |
| ChartSaveTemplates | button | Save As Template |
| ChartAxisTitles | menu | Axis Titles |
| ChartAxes | menu | Axes |
| ChartGridlines | menu | Gridlines |
| ChartFormatSelection | button | Format Selection |
| ChartElementSelector | comboBox | Chart Elements |
| PageMarginsGallery | gallery | Margins |
| DropCapInsertGallery | gallery | Drop Cap |
| TabPictureToolsFormat | tab | Format |
| ShapesInsertGallery | gallery | Shapes |
| ShapeChangeShapeGallery | gallery | Change Shape |
| ShapeFillTextureGallery | gallery | Texture |
| ShapeStylesGallery | gallery | Quick Styles |

| idMso | Control Type | Label |
|---------------------------------|---------------------|--------------------------|
| PageOrientationGallery | gallery | Orientation |
| FileServerTasksMenu | menu | Server |
| FileSendMenu | menu | Send |
| TabInsert | tab | Insert |
| TabReferences | tab | References |
| TabMailings | tab | Mailings |
| TabReviewWord | tab | Review |
| TabView | tab | View |
| GroupFont | group | Font |
| GroupParagraph | group | Paragraph |
| GroupStyles | group | Styles |
| GroupProofing | group | Proofing |
| GroupInsertPages | group | Pages |
| GroupInsertIllustrations | group | Illustrations |
| GroupWordArtText | group | Text |
| GroupParagraphLayout | group | Paragraph |
| GroupCitationsAndBibliography | group | Citations & Bibliography |
| GroupFootnotes | group | Footnotes |
| GroupTableOfContents | group | Table of Contents |
| GroupMailMergeWriteInsertFields | group | Write & Insert Fields |
| GroupMailMergePreviewResults | group | Preview Results |
| GroupMailMergeFinish | group | Finish |
| GroupChangesTracking | group | Tracking |
| GroupComments | group | Comments |
| GroupChanges | group | Changes |
| GroupCompare | group | Compare |
| GroupPictureSize | group | Size |
| GroupTableAlignment | group | Alignment |
| GroupTextBoxText | group | Text |

| idMso | Control Type | Label |
|-----------------------------------|--------------|------------------|
| GroupArrange | group | Arrange |
| GroupShapeStyles | group | Shape Styles |
| GroupChartLayouts | group | Chart Layouts |
| GroupChartStyles | group | Chart Styles |
| GroupChartAxes | group | Axes |
| GroupChartShapes | group | Insert |
| GroupOrganizationChartShapeInsert | group | Insert |
| StylesManageStyles | button | Manage Styles |
| StylesStyleInspector | toggleButton | Style Inspector |
| ObjectEffectPresetGallery | gallery | Preset |
| PictureEffectsPresetGallery | gallery | Preset |
| _3DRotationGallery | gallery | 3-D Rotation |
| TabSmartArtToolsDesign | tab | Design |
| TabSmartArtToolsFormat | tab | Format |
| TabChartToolsDesign | tab | Design |
| TabChartToolsLayout | tab | Layout |
| TabChartToolsFormat | tab | Format |
| ShapeFillColorPicker | gallery | Shape Fill |
| OutlineColorPicker | gallery | Picture Border |
| FileDocumentInspect | button | Inspect Document |
| QuickStylesGallery | gallery | Quick Styles |
| QuickStylesSets | gallery | Style Set |
| ClearFormatting | button | Clear Formatting |
| PanningHand | toggleButton | Panning Hand |
| BulletsGalleryWord | gallery | Bullets |
| NumberingGalleryWord | gallery | Numbering |
| GroupControls | group | Controls |
| GroupZoom | group | Zoom |
| ArrowStyleGallery | gallery | Arrows |

| idMso | Control Type | Label |
|---------------------------------|---------------------|-------------------------------|
| OutlineDashesGallery | gallery | Dashes |
| OutlineWeightGallery | gallery | Weight |
| TabTableToolsLayout | tab | Layout |
| GroupPictureTools | group | Adjust |
| GroupSize | group | Size |
| ReviewTrackChangesMenu | splitButton | Track Changes |
| ReviewAcceptChangeMenu | splitButton | Accept |
| ReviewRejectChangeMenu | splitButton | Reject |
| ReviewBalloonsMenu | menu | Balloons |
| GroupTableRowsAndColumns | group | Rows & Columns |
| GroupTableData | group | Data |
| ObjectAlignMenu | menu | Align |
| ObjectRotateGallery | gallery | Rotate |
| SelectMenu | menu | Select |
| FontColorPicker | gallery | Font Color |
| TableColumnsGallery | gallery | Columns |
| TabHome | tab | Home |
| ChartTitle | gallery | Chart Title |
| ChartPrimaryHorizontalAxisTitle | gallery | Primary Horizontal Axis Title |
| ChartPrimaryVerticalAxisTitle | gallery | Primary Vertical Axis Title |
| ChartDepthAxisTitle | gallery | Depth Axis Title |
| ChartLegend | gallery | Legend |
| ChartDataLabel | gallery | Data Labels |
| ChartPrimaryHorizontalGridlines | gallery | Primary Horizontal Gridlines |
| ChartPrimaryVerticalGridlines | gallery | Primary Vertical Gridlines |
| ChartDepthGridlines | gallery | Depth Gridlines |
| ChartPrimaryHorizontalAxis | gallery | Primary Horizontal Axis |
| ChartPrimaryVerticalAxis | gallery | Primary Vertical Axis |

| idMso | Control Type | Label |
|-------------------------------|---------------------|-----------------|
| ChartDepthAxis | gallery | Depth Axis |
| ChartDataTable | gallery | Data Table |
| ChartTrendline | gallery | Trendline |
| ChartErrorBars | gallery | Error Bars |
| ChartLines | gallery | Lines |
| ChartUpDownBars | gallery | Up/Down Bars |
| ChartPlotArea | gallery | Plot Area |
| ChartWall | gallery | Chart Wall |
| ChartFloor | gallery | Chart Floor |
| TabPageLayoutWord | tab | Page Layout |
| SmartArtAddShape | button | Add Shape |
| SmartArtLargerShape | button | Larger |
| SmartArtSmallerShape | button | Smaller |
| SmartArtResetGraphic | button | Reset Graphic |
| SmartArtTextPane | toggleButton | Text Pane |
| SmartArtEditIn2D | toggleButton | Edit in 2-D |
| SmartArtLayoutGallery | gallery | Change Layout |
| SmartArtMoreLayoutsDialog | button | More Layouts... |
| SmartArtStylesGallery | gallery | Quick Styles |
| SmartArtChangeColorsGallery | gallery | Change Colors |
| ObjectEffectSoftEdgesGallery | gallery | Soft Edges |
| ObjectEffectGlowGallery | gallery | Glow |
| GradientGallery | gallery | Gradient |
| ObjectEffectShadowGallery | gallery | Shadow |
| TextEffectTransformGallery | gallery | Transform |
| TabHeaderAndFooterToolsDesign | tab | Design |
| GroupHeaderFooterOptions | group | Options |
| ReviewPreviousCommentWord | button | Previous |
| ReviewNextCommentWord | button | Next |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|---|
| BulletDefineNew | button | Define New Bullet... |
| DefineNewNumberFormat | button | Define New Number Format... |
| ThemeColorsGallery | gallery | Colors |
| HeaderInsertGallery | gallery | Header |
| FooterInsertGallery | gallery | Footer |
| CoverPageInsertGallery | gallery | Cover Page |
| PageNumberFieldInsertGallery | gallery | Current Position |
| WatermarkGallery | gallery | Watermark |
| EquationInsertGallery | gallery | Equation |
| QuickTablesInsertGallery | gallery | Quick Tables |
| QuickPartsInsertGallery | gallery | Quick Parts |
| GroupSmartArtLayouts | group | Layouts |
| GroupSmartArtQuickStyles | group | SmartArt Styles |
| GroupSmartArtCreateGraphic | group | Create Graphic |
| GroupSmartArtReset | group | Reset |
| GroupSmartArtSize | group | Size |
| SaveSelectionToQuickPartGallery | button | Save Selection to Quick Part Gallery... |
| SaveSelectionToCoverPageGallery | button | Save Selection to Cover Page Gallery... |
| SaveSelectionToEquationGallery | button | Save Selection to Equation Gallery... |
| SaveSelectionToFooterGallery | button | Save Selection to Footer Gallery... |
| SaveSelectionToHeaderGallery | button | Save Selection to Header Gallery... |
| SaveSelectionToPageNumberGallery | button | Save Selection to Page Number Gallery... |
| SaveSelectionToQuickTablesGallery | button | Save Selection to Quick Tables Gallery... |
| SaveSelectionToWaterMarkGallery | button | Save Selection to Watermark Gallery... |
| ThemeSearchOfficeOnline | button | More Themes on Microsoft |

| idMso | Control Type | Label |
|--------------------------------|--------------|-----------------------|
| | | Office Online... |
| TabAddIns | tab | Add-Ins |
| ReviewShowMarkupMenu | menu | Show Markup |
| ObjectEditShapeMenu | menu | Edit Shape |
| SymbolInsertGallery | gallery | Symbol |
| TableStyleClear | button | Clear |
| FileSaveAsPdfOrXps | button | Publish as PDF or XPS |
| FileSaveAsWordOpenDocumentText | button | OpenDocument Text |
| SearchLibraries | button | Search Libraries... |
| MoreControlsDialog | button | More Controls... |
| GroupCode | group | Code |
| TabDeveloper | tab | Developer |
| GroupXml | group | XML |
| XmlStructure | toggleButton | Structure |
| XmlSchema | button | Schema |
| XmlExpansionPacksWord | button | Expansion Packs |
| GroupCaptions | group | Captions |
| GroupIndex | group | Index |
| GroupTableOfAuthorities | group | Table of Authorities |
| GroupEditing | group | Editing |
| SelectMenuExcel | splitButton | Find & Select |
| GroupClipboard | group | Clipboard |
| GroupInsertTables | group | Tables |
| GroupInsertLinks | group | Links |
| GroupInsertSymbols | group | Symbols |
| GroupInsertBarcode | group | Barcode |
| PageSizeGallery | gallery | Size |
| ObjectPictureFill | button | Picture... |
| TextWrappingMenu | menu | Text Wrapping |
| WindowSwitchWindowsMenuWord | menu | Switch Windows |

| idMso | Control Type | Label |
|--------------------------------------|--------------|------------------------------|
| ThemeColorsCreateNew | button | Create New Theme Colors... |
| ThemeFontsCreateNew | button | Create New Theme Fonts... |
| ShapeFillMoreGradientsDialog | button | More Gradients... |
| ShadowOptionsDialog | button | Shadow Options... |
| BuildingBlocksOrganizer | button | Building Blocks Organizer... |
| ReviewCompareMenu | menu | Compare |
| ReviewCompareTwoVersions | button | Compare... |
| ReviewCombineRevisions | button | Combine... |
| ReviewCompareMajorVersion | button | Major Version |
| ReviewCompareLastVersion | button | Last Version |
| ReviewCompareSpecificVersion | button | Specific Version... |
| PropertyInsert | gallery | Document Property |
| ObjectsAlignSelectedSmart | toggleButton | Align Selected Objects |
| ObjectsAlignRelativeToContainerSmart | toggleButton | Align to Slide |
| ObjectsAlignLeftSmart | button | Align Left |
| ObjectsAlignRightSmart | button | Align Right |
| ObjectsAlignTopSmart | button | Align Top |
| ObjectsAlignBottomSmart | button | Align Bottom |
| ObjectsAlignCenterHorizontalSmart | button | Align Center |
| ObjectsAlignMiddleVerticalSmart | button | Align Middle |
| AlignDistributeHorizontally | button | Distribute Horizontally |
| AlignDistributeVertically | button | Distribute Vertically |
| GroupProtect | group | Protect |
| IndexUpdate | button | Update Index |
| MarginsCustomMargins | button | Custom Margins... |
| MailMergeFinishAndMergeMenu | menu | Finish & Merge |
| HyphenationMenu | menu | Hyphenation |
| HyphenationAutomatic | toggleButton | Automatic |

| idMso | Control Type | Label |
|-------------------------------|---------------------|--------------------------------|
| HyphenationNone | toggleButton | None |
| MailMergeRules | menu | Rules |
| TabWordArtToolsFormat | tab | Format |
| MailMergeStartMailMergeMenu | menu | Start Mail Merge |
| MailMergeStartLetters | toggleButton | Letters |
| MailMergeStartEmail | toggleButton | E-Mail Messages |
| MailMergeStartEnvelopes | toggleButton | Envelopes... |
| MailMergeStartLabels | toggleButton | Labels... |
| MailMergeStartDirectory | toggleButton | Directory |
| MailMergeClearMergeType | toggleButton | Normal Word Document |
| MailMergeSelectRecipients | menu | Select Recipients |
| TableOfContentsAddTextGallery | gallery | Add Text |
| BorderColorPicker | gallery | Pen Color |
| TranslationToolTip | gallery | Translation ScreenTip |
| ThemeFontsGallery | gallery | Fonts |
| ThemeEffectsGallery | gallery | Effects |
| FileProperties | toggleButton | Properties |
| TabPrintPreview | tab | Print Preview |
| GroupPrintPreviewPrint | group | Print |
| GroupPrintPreviewPreview | group | Preview |
| BreaksGallery | gallery | Breaks |
| LineNumbersMenu | menu | Line Numbers |
| LineNumbersOff | toggleButton | None |
| LineNumbersContinuous | toggleButton | Continuous |
| LineNumbersResetPage | toggleButton | Restart Each Page |
| LineNumbersResetSection | toggleButton | Restart Each Section |
| LineNumbersSuppress | toggleButton | Suppress for Current Paragraph |
| TabOutlining | tab | Outlining |
| GroupOutliningClose | group | Close |

| idMso | Control Type | Label |
|--------------------------------------|--------------|--|
| GroupOutliningTools | group | Outline Tools |
| GroupMasterDocument | group | Master Document |
| TableSelectMenu | menu | Select |
| TableDeleteRowsAndColumnsMenuWord | menu | Delete |
| GroupTableMerge | group | Merge |
| TableAutoFitMenu | menu | AutoFit |
| GroupTableDrawBorders | group | Draw Borders |
| TableBordersMenu | splitButton | Borders |
| FileCreateDocumentWorkspace | toggleButton | Create Document Workspace |
| FileSaveToDocumentManagementServer | button | Document Management Server |
| FileDocumentManagementInformation | toggleButton | Document Management Information |
| QuickAccessToolbarCustomization | button | Customize Quick Access Toolbar... |
| FilePrepareMenu | menu | Prepare |
| FileMarkAsFinal | toggleButton | Mark as Final |
| FileAddDigitalSignature | button | Add a Digital Signature |
| SignatureServicesAdd | button | Add Signature Services... |
| QuickStylesSaveSelectionAsNew | button | Save Selection as a New Quick Style... |
| StylesPaneNewStyle | button | New Style... |
| QuickStylesSaveQuickStyleSet | button | Save as Quick Style Set... |
| ChangeCaseGallery | gallery | Change Case |
| AlignJustifyMenu | menu | Justify |
| ControlProperties | button | Properties |
| GroupHeaderFooterInsert | group | Insert |
| GroupHeaderFooterNavigation | group | Navigation |
| GroupHeaderFooterPosition | group | Position |
| HeaderFooterDifferentFirstPageWord | checkBox | Different First Page |
| HeaderFooterDifferentOddEvenPageWord | checkBox | Different Odd & Even |

| idMso | Control Type | Label |
|------------------------------------|--------------|-------------------------------|
| | | Pages |
| HeaderFooterShowDocumentText | checkBox | Show Document Text |
| BlankPageInsert | button | Blank Page |
| ShadowStyleGalleryClassic | gallery | Shadow Effects |
| TabOrganizationChartToolsFormat | tab | Format |
| TabDiagramToolsFormatClassic | tab | Format |
| GroupInkSelect | group | Select |
| WordArtSpacingMenu | menu | Spacing |
| TextAlignMenu | menu | Alignment |
| DiagramChangeToMenuClassic | menu | Change To |
| PictureBrightnessGallery | gallery | Brightness |
| PictureContrastGallery | gallery | Contrast |
| PicturePositionGallery | gallery | Position |
| TabPictureToolsFormatClassic | tab | Format |
| PictureBrightnessAndContrastDialog | button | Picture Correction Options... |
| GroupMailMergeStart | group | Start Mail Merge |
| OleObjectInsertMenu | splitButton | Object |
| ShadingColorPicker | gallery | Shading |
| ShadingColorsMoreColorsDialog | button | More Colors... |
| SmartArtAddShapeAfter | button | Add Shape After |
| SmartArtAddShapeBefore | button | Add Shape Before |
| SmartArtAddShapeAbove | button | Add Shape Above |
| SmartArtAddShapeBelow | button | Add Shape Below |
| SmartArtAddAssistant | button | Add Assistant |
| ChartSwitchRowColumn | button | Switch Row/Column |
| ChartShowData | button | Edit Data... |
| ChartRefresh | button | Refresh Data |
| ChartChangeType | button | Change Chart Type... |
| GroupChartData | group | Data |

| idMso | Control Type | Label |
|---------------------------------------|--------------|-------------------------------|
| GroupChartType | group | Type |
| _3DRotationOptionsDialog | button | 3-D Rotation Options... |
| _3DBevelOptionsDialog | button | 3-D Options... |
| SmartArtOrganizationChartLeftHanging | button | Left Hanging |
| SmartArtOrganizationChartRightHanging | button | Right Hanging |
| SmartArtOrganizationChartBoth | button | Both |
| SmartArtOrganizationChartStandard | button | Standard |
| SmartArtRightToLeft | toggleButton | Right to Left |
| ListLevelGallery | gallery | Change List Level |
| MultilevelListGallery | gallery | Multilevel List |
| ListDefineNew | button | Define New Multilevel List... |
| ListDefineNewStyle | button | Define New List Style... |
| ViewMessageBar | checkBox | Message Bar |
| ApplyStylesPane | toggleButton | Apply Styles... |
| InsertAlignmentTab | button | Insert Alignment Tab |
| ShapeStylesOtherThemeFillsGallery | gallery | Other Theme Fills |
| SmartArtOrganizationChartMenu | menu | Layout |
| ReviewChangeTrackingOptions | button | Change Tracking Options... |
| SymbolsDialog | button | More Symbols... |
| ReviewReviewingPaneHorizontal | toggleButton | Reviewing Pane Horizontal... |
| ReviewReviewingPaneVertical | toggleButton | Reviewing Pane Vertical... |
| _3DEffectsGalleryClassic | gallery | 3-D Effects |
| _3DDirectionGalleryClassic | gallery | Direction |
| _3DLightingGalleryClassic | gallery | Lighting |
| TabDrawingToolsFormatClassic | tab | Format |
| GroupShadowEffects | group | Shadow Effects |
| Group3DEffects | group | 3-D Effects |
| WordArtStylesGalleryClassic | gallery | WordArt |

| idMso | Control Type | Label |
|---------------------------------|---------------------|----------------------|
| WordArtInsertGalleryClassic | gallery | WordArt |
| TableInsertGallery | gallery | Table |
| ShapeStylesGalleryClassic | gallery | Shape Styles |
| WordArtChangeShapeGallery | gallery | Change Shape |
| ShadowColorPickerClassic | gallery | Shadow Color |
| _3DEffectColorPickerClassic | gallery | 3-D Color |
| ShapeFillGradientGalleryClassic | gallery | Gradient |
| AsianLayoutMenu | menu | Asian Layout |
| JapaneseGreetingsInsertMenu | menu | Japanese Greetings |
| AlignJustifyWithMixedLanguages | toggleButton | Justify |
| AlignJustifyLow | toggleButton | Justify Low |
| AlignJustifyMedium | toggleButton | Justify Medium |
| AlignJustifyHigh | toggleButton | Justify High |
| AlignJustifyThai | toggleButton | Distribute |
| TextHighlightColorPicker | gallery | Text Highlight Color |
| PageColorPicker | gallery | Page Color |
| GoToHeader | button | Go to Header |
| GoToFooter | button | Go to Footer |
| HighlightingStop | button | Stop Highlighting |
| UnderlineGallery | gallery | Underline |
| UnderlineColorPicker | gallery | Underline Color |
| UnderlineMoreUnderlinesDialog | button | More Underlines... |
| TextDirectionGalleryWord | gallery | Text Direction |
| GroupAddInsMenuCommands | group | Menu Commands |
| GroupAddInsToolbarCommands | group | Toolbar Commands |
| GroupInk | group | Ink |
| TabInkToolsPens | tab | Pens |
| GroupInkPens | group | Pens |
| GroupInkClose | group | Close |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|--|
| InkBallpointPen | toggleButton | Ballpoint Pen |
| InkFeltTipPen | toggleButton | Felt Tip Pen |
| InkHighlighter | toggleButton | Highlighter |
| GroupBorder | group | Border |
| PictureRecolorGalleryWord | gallery | Recolor |
| _3DSurfaceMaterialGalleryClassic | gallery | Surface |
| _3DExtrusionDepthGalleryClassic | gallery | Depth |
| GroupHeaderFooter | group | Header & Footer |
| GroupPageLayoutSetup | group | Page Setup |
| GroupPageBackground | group | Page Background |
| ThemeSaveCurrent | button | Save Current Theme... |
| ThemesGallery | gallery | Themes |
| ChartResetToMatchStyle | button | Reset to Match Style |
| Chart3DView | button | 3-D Rotation... |
| ObjectSizeDialog | button | Size... |
| AutoTextGallery | gallery | AutoText |
| TextBoxInsertGallery | gallery | Text Box |
| PageNumbersInHeaderInsertGallery | gallery | Top of Page |
| PageNumbersInFooterInsertGallery | gallery | Bottom of Page |
| PageNumbersInMarginsInsertGallery | gallery | Page Margins |
| SaveSelectionToAutoTextGallery | button | Save Selection to AutoText Gallery |
| SaveSelectionToTextBoxGallery | button | Save Selection to Text Box Gallery |
| SaveSelectionToPageNumberTop | button | Save Selection as Page Number (Top) |
| SaveSelectionToPageNumberBottom | button | Save Selection as Page Number (Bottom) |
| SaveSelectionToPageNumberMargin | button | Save Selection as Page Number (Margin) |
| HeaderFooterEditHeader | button | Edit Header |
| TableStyleHeaderRowWord | checkBox | Header Row |

| idMso | Control Type | Label |
|--|--------------|---|
| TableStyleTotalRowWord | checkBox | Total Row |
| TableStylesFirstColumnWord | checkBox | First Column |
| TableStyleLastColumnWord | checkBox | Last Column |
| TableStyleBandedRowsWord | checkBox | Banded Rows |
| TableStyleBandedColumnsWord | checkBox | Banded Columns |
| TableStyleModify | button | Modify Table Style... |
| TabTableToolsDesign | tab | Design |
| TableStylesGalleryWord | gallery | Table Styles |
| ReviewViewChangesInTheSourceDocument | button | View changes in the source document |
| GroupThemesWord | group | Themes |
| LayoutOptionsDialog | button | More Layout Options... |
| DrawingObjectFormatDialog | button | Advanced Tools... |
| ReflectionGallery | gallery | Reflection |
| PictureRecolorGallery | gallery | Recolor |
| SmartArtPromote | button | Promote |
| SmartArtDemote | button | Demote |
| ChartTitleOptionsDialog | button | More Title Options... |
| ChartLegendOptionsDialogDialog | button | More Legend Options... |
| ChartDataLabelDialog | button | More Data Label Options... |
| ChartPrimaryHorizontalAxisTitleOptionsDialog | button | More Primary Horizontal Axis Title Options... |
| ChartPrimaryVerticalAxisTitleOptionsDialog | button | More Primary Vertical Axis Title Options... |
| ChartSecondaryHorizontalAxisTitleOptionsDialog | button | More Secondary Horizontal Axis Title Options... |
| ChartSecondaryVerticalAxisTitleOptionsDialog | button | More Secondary Vertical Axis Title Options... |
| ChartDepthAxisTitleOptionsDialog | button | More Depth Axis Title Options... |
| ChartPrimaryHorizontalGridlinesOptionsDialog | button | More Primary Horizontal Gridlines Options... |
| ChartPrimaryVerticalGridlinesOptionsDialog | button | More Primary Vertical |

| idMso | Control Type | Label |
|--|--------------|--|
| | | Gridlines Options... |
| ChartSecondaryHorizontalGridlinesOptionsDialog | button | More Secondary Horizontal Gridlines Options... |
| ChartSecondaryVerticalGridlinesOptionsDialog | button | More Secondary Vertical Gridlines Options... |
| ChartDepthGridlinesOptionsDialog | button | More Depth Gridlines Options... |
| ChartPrimaryHorizontalAxisOptionsDialog | button | More Primary Horizontal Axis Options... |
| ChartPrimaryVerticalAxisOptionsDialog | button | More Primary Vertical Axis Options... |
| ChartSecondaryHorizontalAxisOption | button | More Secondary Horizontal Axis Options... |
| ChartSecondaryVerticalAxisOptionsDialog | button | More Secondary Vertical Axis Options... |
| ChartDepthAxisOptionsDialog | button | More Depth Axis Options... |
| ChartDataTableOptionsDialog | button | More Data Table Options... |
| ChartTrendlineOptionsDialog | button | More Trendline Options... |
| ChartErrorBarsOptionsDialog | button | More Error Bars Options... |
| ChartUpDownBarsOptionsDialog | button | More Up/Down Bars Options... |
| ChartPlotAreaOptionsDialog | button | More Plot Area Options... |
| ChartWallOptionsDialog | button | More Walls Options... |
| ChartFloorOptionsDialog | button | More Floor Options... |
| ChartSecondaryHorizontalAxisTitle | gallery | Secondary Horizontal Axis Title |
| ChartSecondaryVerticalAxisTitle | gallery | Secondary Vertical Axis Title |
| ChartSecondaryHorizontalGridlines | gallery | Secondary Horizontal Gridlines |
| ChartSecondaryVerticalGridlines | gallery | Secondary Vertical Gridlines |
| ChartSecondaryHorizontalAxis | gallery | Secondary Horizontal Axis |
| ChartSecondaryVerticalAxis | gallery | Secondary Vertical Axis |
| GroupAddInsCustomToolbars | group | Custom Toolbars |
| ReviewDeleteCommentsMenu | splitButton | Delete |

| idMso | Control Type | Label |
|---------------------------------------|---------------------|--|
| ObjectBringToFrontMenu | splitButton | Bring to Front |
| ObjectSendToBackMenu | splitButton | Send to Back |
| ObjectsGroupMenu | menu | Group |
| SignatureLineInsertMenu | splitButton | Signature Line |
| FileSaveAsWord97_2003 | button | Word 97-2003 Document |
| EnglishWritingAssistant | button | English Assistant |
| TableOfContentsGallery | gallery | Table of Contents |
| SaveSelectionToTableOfContentsGallery | button | Save Selection to Table of Contents Gallery... |
| ObjectsAlignRelativeToMargin | toggleButton | Align to Margin |
| TabTextBoxToolsFormat | tab | Format |
| TextBoxStyleGallery | gallery | Text Box Style |
| TextBoxPositionGallery | gallery | Position |
| FileSaveAsMenu | splitButton | Save As Other Format |
| FilePrintMenu | splitButton | Preview and Print |
| FilePermissionRestrictMenu | menu | Restrict Permission |
| GroupEnvelopeLabelCreate | group | Create |
| ReviewInkCommentNew | button | Ink Comment |
| DocumentPanelTemplate | button | Document Panel |
| ObjectSizeDialogClassic | button | Size... |
| TextBoxInsertVerticalWord | button | Draw Vertical Text Box |
| BevelShapeGallery | gallery | Bevel |
| _3DBevelPictureTopGallery | gallery | Bevel |
| BibliographyAddNewPlaceholder | button | Add New Placeholder... |
| GroupTable | group | Table |
| GroupTableCellSize | group | Cell Size |
| GlowColorPicker | gallery | More Glow Colors |
| RecolorColorPicker | gallery | More Variations |
| GlowColorMoreColorsDialog | button | More Colors... |
| PictureRecolorMoreColorsDialog | button | More Colors... |

| idMso | Control Type | Label |
|----------------------------------|---------------------|-------------------------------------|
| QuickStylesResetDocumentStyles | button | Reset Document Quick Styles |
| QuickStylesResetFromTemplate | button | Reset to Quick Styles from Template |
| WatermarkRemove | button | Remove Watermark |
| CoverPageRemove | button | Remove Current Cover Page |
| HeaderFooterRemoveHeaderWord | button | Remove Header |
| HeaderFooterRemoveFooterWord | button | Remove Footer |
| PageNumbersRemove | button | Remove Page Numbers |
| TableOfContentsRemove | button | Remove Table of Contents |
| SmartArtAddBullet | button | Add Bullet |
| ThemeResetFromTemplate | button | Reset to Theme from Template |
| PictureChange | button | Change Picture... |
| GroupWordArtStyles | group | WordArt Styles |
| TextFillColorPicker | gallery | Text Fill |
| TextOutlineColorPicker | gallery | Text Outline |
| TextOutlineColorMoreColorsDialog | button | More Outline Colors... |
| TextEffectsMenu | menu | Text Effects |
| TextStylesGallery | gallery | Quick Styles |
| WordArtClear | button | Clear WordArt |
| TextPictureFill | button | Picture... |
| TextFillGradientGallery | gallery | Gradient |
| TextFillMoreGradientsDialog | button | More Gradients... |
| TextFillTextureGallery | gallery | Texture |
| TextOutlineDashesGallery | gallery | Dashes |
| TextOutlineMoreLinesDialog | button | More Lines... |
| TextOutlineWeightGallery | gallery | Weight |
| TextEffectShadowGallery | gallery | Shadow |
| TextEffectsMoreShadowsDialog | button | Shadow Options... |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|--------------------------------|
| TextEffectsBevelMore3DOptionsDialog | button | 3-D Options... |
| TextEffects3DRotationGallery | gallery | 3-D Rotation |
| TextEffects3DRotationOptionsDialog | button | 3-D Rotation Options... |
| TextEffectGlowGallery | gallery | Glow |
| TextGlowColorPicker | gallery | More Glow Colors |
| TextGlowColorMoreColorsDialog | button | More Colors... |
| TextReflectionGallery | gallery | Reflection |
| ShapeEffectsMenu | menu | Shape Effects |
| UpgradeDocument | button | Convert |
| GroupHeaderFooterClose | group | Close |
| GroupChartCurrentSelection | group | Current Selection |
| GroupChartLabels | group | Labels |
| PageSizeMorePaperSizesDialog | button | More Paper Sizes... |
| LineNumbersOptionsDialog | button | Line Numbering Options... |
| TableOfAuthoritiesUpdate | button | Update Table |
| TableOfFiguresUpdate | button | Update Table |
| ContentControlsGroup | button | Group |
| BevelTextGallery | gallery | Bevel |
| PictureCorrectionsDialog | button | Picture Corrections Options... |
| GroupTextBoxStyles | group | Text Box Styles |
| GroupTableStylesWord | group | Table Styles |
| GroupWordArtStylesClassic | group | WordArt Styles |
| PageBorderAndShadingDialog | button | Page Borders... |
| OutlineViewClose | button | Close Outline View |
| SmartArtAddShapeSplitMenu | splitButton | Add Shape Options |
| ContentControlRichText | button | Rich Text |
| ContentControlText | button | Text |
| ContentControlPicture | button | Picture |
| ContentControlComboBox | button | Combo Box |

| idMso | Control Type | Label |
|---|---------------------|------------------------|
| ContentControlDropDownList | button | Drop-Down List |
| ContentControlBuildingBlockGallery | button | Building Block Gallery |
| ContentControlDate | button | Date |
| AutoSummaryToolsMenu | menu | AutoSummary Tools |
| GroupInkFormat | group | Format |
| InkColorPicker | gallery | Color |
| GroupDiagramLayoutClassic | group | Layout |
| GroupDiagramStylesClassic | group | Styles |
| GroupOrganizationChartLayoutClassic | group | Layout |
| GroupOrganizationChartStyleClassic | group | Styles |
| TextBoxDrawMenu | menu | Draw Text Box |
| TextBoxInsertWord | button | Text Box |
| GroupOrganizationChartSelect | group | Select |
| ShapeFillEffectMoreGradientsDialogClassic | button | More Gradients... |
| ShapeFillEffectMoreTexturesDialogClassic | button | More Textures... |
| ShapeFillEffectPatternClassic | button | Pattern... |
| BibliographyGallery | gallery | Bibliography |
| CustomHeaderGallery | gallery | Custom Header |
| CustomFooterGallery | gallery | Custom Footer |
| CustomCoverPageGallery | gallery | Custom Cover Page |
| CustomPageNumberGallery | gallery | Custom Page Number |
| CustomPageNumberTopGallery | gallery | Custom Top of Page |
| CustomPageNumberBottomGallery | gallery | Custom Bottom of Page |
| CustomPageMargins | gallery | Custom Page Margins |
| CustomWatermarkGallery | gallery | Custom Watermark |
| CustomEquationsGallery | gallery | Custom Equation |
| CustomTablesGallery | gallery | Custom Tables |
| CustomQuickPartsGallery | gallery | Custom Quick Parts |
| CustomAutoTextGallery | gallery | Custom AutoText |

| idMso | Control Type | Label |
|--|--------------|---|
| CustomTextBoxGallery | gallery | Custom Text Box |
| CustomTableOfContentsGallery | gallery | Custom Table of Contents |
| CustomBibliographyGallery | gallery | Custom Bibliography |
| CustomGallery1 | gallery | Custom Gallery 1 |
| CustomGallery2 | gallery | Custom Gallery 2 |
| CustomGallery3 | gallery | Custom Gallery 3 |
| CustomGallery4 | gallery | Custom Gallery 4 |
| CustomGallery5 | gallery | Custom Gallery 5 |
| SaveSelectionToBibliographyGallery | button | Save Selection to Bibliography Gallery... |
| MailMergeReceipientsUseOutlookContacts | button | Select from Outlook Contacts... |
| FootnoteNext | splitButton | Next Footnote |
| ReviewReviewingPaneMenu | splitButton | Reviewing Pane |
| GroupSizeClassic | group | Size |
| GroupPictureSizeClassic | group | Size |
| GroupPictureToolsClassic | group | Adjust |
| GalleryAllShapesAndCanvas | gallery | Shapes |
| GroupShapesClassic | group | Insert Shapes |
| GroupSmartArtShapes | group | Shapes |
| GroupShapeStylesClassic | group | Shape Styles |
| GroupInsertText | group | Text |
| Drawing1ColorPickerFill | gallery | Shape Fill |
| ShapeOutlineColorPicker | gallery | Picture Border |
| Drawing1ColorPickerLineStyles | gallery | Picture Border |
| Drawing1GalleryTextures | gallery | Texture |
| InsertBuildingBlocksEquationsGallery | gallery | Equation |
| Drawing1GalleryBrightness | gallery | Brightness |
| Drawing1GalleryContrast | gallery | Contrast |
| GroupDiagramArrangeClassic | group | Arrange |

| idMso | Control Type | Label |
|-------------------------------|--------------|-----------------------|
| GroupTextBoxArrange | group | Arrange |
| ContentControlsGroupMenu | menu | Group |
| ContentControlsUngroup | button | Ungroup |
| ControlsGalleryClassic | gallery | Legacy Tools |
| ReviewShowSourceDocumentsMenu | gallery | Show Source Documents |
| HeaderFooterEditFooter | button | Edit Footer |
| QuickStylesSetAsDefault | button | Set as Default |
| EquationInsertNew | button | Insert New Equation |
| EquationProfessional | button | Professional |
| EquationLinearFormat | button | Linear |
| EquationNormalText | toggleButton | Normal Text |
| EquationSymbolsInsertGallery | gallery | Equation Symbols |
| EquationIntegralGallery | gallery | Integral |
| EquationFractionGallery | gallery | Fraction |
| EquationRadicalGallery | gallery | Radical |
| EquationLargeOperatorGallery | gallery | Large Operator |
| EquationDelimiterGallery | gallery | Bracket |
| EquationScriptGallery | gallery | Script |
| EquationFunctionGallery | gallery | Function |
| EquationAccentGallery | gallery | Accent |
| EquationLimitGallery | gallery | Limit and Log |
| EquationOperatorGallery | gallery | Operator |
| EquationMatrixGallery | gallery | Matrix |
| EquationOptions | button | Equation Options... |
| TabEquationToolsDesign | tab | Design |
| GroupEquationTools | group | Tools |
| GroupEquationSymbols | group | Symbols |
| GroupEquationStructures | group | Structures |
| MailMergeMergeFieldInsertMenu | splitButton | Insert Merge Field |

| idMso | Control Type | Label |
|---------------------------------|---------------------|------------------------------|
| PasteMenu | splitButton | Paste |
| GroupPictureStyles | group | Picture Styles |
| PictureStylesGallery | gallery | Quick Styles |
| ReviewAcceptChangeAndMoveToNext | button | Accept and Move to Next |
| ReviewRejectChangeAndMoveToNext | button | Reject and Move to Next |
| PictureEffectsShadowGallery | gallery | Shadow |
| PictureEffectsGlowGallery | gallery | Glow |
| PictureEffectsSoftEdgesGallery | gallery | Soft Edges |
| PictureReflectionGallery | gallery | Reflection |
| PictureRotationGallery | gallery | 3-D Rotation |
| InkToolsClose | button | Close Ink Tools |
| GroupChineseTranslation | group | Chinese Translation |
| LineSpacingMenu | menu | Line spacing |
| GroupDocumentViews | group | Document Views |
| GroupViewShowHide | group | Show/Hide |
| GroupWindow | group | Window |
| ViewGridlines | checkBox | View Gridlines |
| FileDocumentEncrypt | toggleButton | Encrypt Document |
| WordArtFormatDialog | button | Format Text Effects... |
| ObjectRotationOptionsDialog | button | More Rotation Options... |
| MoreTextureOptions | button | More Textures... |
| TextFillColorMoreColorsDialog | button | More Fill Colors... |
| ZoomTwoPages | button | Two Pages |
| QuickPartsInsertFromOnline | button | Get More on Office Online... |
| GroupPrintPreviewPageSetup | group | Page Setup |
| ShowRuler | checkBox | Ruler |
| FileEmailAsPdfEmailAttachment | button | E-mail as PDF Attachment |
| FileEmailAsXpsEmailAttachment | button | E-mail as XPS Attachment |
| GroupTemplates | group | Templates |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|---------------------------|
| SpellingMenu | splitButton | Spelling |
| PictureEffectsMenu | menu | Picture Effects |
| PictureShapeGallery | gallery | Change Shape |
| PictureBorderColorPickerClassic | gallery | Picture Border |
| GroupChartBackground | group | Background |
| GroupChartAnalysis | group | Analysis |
| FileNewBlogPost | button | Blog |
| TabBlogPost | tab | Blog Post |
| GroupBlogPublish | group | Blog |
| BlogPublishMenu | splitButton | Publish |
| BlogPublish | button | Publish |
| BlogPublishDraft | button | Publish as Draft |
| BlogManageAccounts | button | Manage Accounts |
| BlogCategoryInsert | button | Insert Category |
| BlogOpenExisting | button | Open Existing |
| GroupBlogBasicText | group | Basic Text |
| TabBlogInsert | tab | Insert |
| GroupBlogInsertText | group | Text |
| NewTableStyleWord | button | New Table Style... |
| MenuPublish | menu | Publish |
| ChangeStylesMenu | menu | Change Styles |
| GroupBlogInsertLinks | group | Links |
| FileCompatibilityCheckerWord | button | Run Compatibility Checker |
| FileSaveAsOtherFormats | button | Other Formats |
| FileSaveAsWordDocx | button | Word Document |
| FileSaveAsWordDotx | button | Word Template |
| ZoomCurrent100 | button | 100% |
| Drawing1ColorPickerLineStyleWordArt | gallery | Picture Border |
| Drawing1ColorPickerFillWordArt | gallery | Shape Fill |

| idMso | Control Type | Label |
|--------------------------------|---------------------|-------------------------------------|
| BlogInsertCategories | button | Insert Category |
| GroupTableLayout | group | Table Style Options |
| TextFillMoreTextures | button | More Textures... |
| GroupMacros | group | Macros |
| PlayMacro | button | Macros |
| MenuMacros | splitButton | Macros |
| AdvertisePublishAs | button | Find add-ins for other file formats |
| ReviewProtectDocumentMenu | menu | Protect Document |
| ReviewRestrictFormatting | toggleButton | Restrict Formatting and Editing |
| BlogHomePage | button | Home Page |
| GroupBlogProofing | group | Proofing |
| GroupBlogStyles | group | Styles |
| AlternativeText | button | Size... |
| ThemeBrowseForThemes | button | Browse for Themes... |
| FileCheckOutDiscard | button | Discard Check Out |
| GroupBlogSymbols | group | Symbols |
| ClearMenuWord | menu | Clear |
| MdiChildSystemMenu | menu | Document |
| NudgeMenu | menu | Nudge |
| RevisionsMenu | menu | Track Changes |
| TableCellVerticalAlignmentMenu | menu | Alignment |
| FramesetMenu | menu | Frames |

3.1.2 Excel 2007

| idMso | Control Type | Label |
|--------------|---------------------|--------------|
| Spelling | button | Spelling... |
| FileSave | button | Save |
| FilePrint | button | Print |

| idMso | Control Type | Label |
|-----------------------|---------------------|------------------|
| ChartInsert | button | Chart... |
| FileNew | button | New |
| Copy | button | Copy |
| Cut | button | Cut |
| Paste | button | Paste |
| FileOpen | button | Open |
| ZoomPrintPreviewExcel | toggleButton | Zoom |
| Repeat | button | Repeat |
| UnderlineDouble | toggleButton | Double Underline |
| FileClose | button | Close |
| FormatPainter | toggleButton | Format Painter |
| FilePrintPreview | button | Print Preview |
| Bold | toggleButton | Bold |
| Italic | toggleButton | Italic |
| Underline | toggleButton | Underline |
| DarkShading | button | Dark Shading |
| AlignLeft | toggleButton | Align Left |
| AlignRight | toggleButton | Align Right |
| AlignCenter | toggleButton | Center |
| AlignJustify | toggleButton | Justify |
| Undo | gallery | Undo |
| Redo | gallery | Redo |
| BorderTop | toggleButton | Top Border |
| BorderBottom | toggleButton | Bottom Border |
| BorderLeft | toggleButton | Left Border |
| BorderRight | toggleButton | Right Border |
| BorderInside | button | Inside Borders |
| BorderOutside | button | Outside Borders |
| BorderNone | button | No Border |

| idMso | Control Type | Label |
|----------------------|---------------------|------------------|
| ObjectsGroup | button | Group |
| ObjectsUngroup | button | Ungroup |
| ObjectBringToFront | button | Bring to Front |
| ObjectSendToBack | button | Send to Back |
| ObjectBringForward | button | Bring Forward |
| ObjectSendBackward | button | Send Backward |
| ViewFullScreenView | toggleButton | Full Screen |
| ObjectsSelect | toggleButton | Select Objects |
| MacroRecord | button | Record Macro... |
| MacroPlay | button | Macros |
| ObjectFlipHorizontal | button | Flip Horizontal |
| ObjectFlipVertical | button | Flip Vertical |
| ObjectRotateRight90 | button | Rotate Right 90° |
| ObjectRotateLeft90 | button | Rotate Left 90° |
| ShapeFreeform | toggleButton | Freeform |
| ObjectEditPoints | toggleButton | Edit Points |
| FormControlEditBox | button | Edit Box |
| FormControlCheckBox | button | Check Box |
| FormControlComboBox | button | Combo Box |
| PropertySheet | button | Property Sheet |
| Lock | toggleButton | Lock |
| AutoSum | button | Sum |
| StylesDialogClassic | button | Edit Cell Styles |
| Camera | button | Camera |
| FormControlButton | button | Button |
| Calculator | button | Calculator |
| Strikethrough | toggleButton | Strikethrough |
| CellsDelete | button | Delete Cells... |
| CellsInsertDialog | button | Insert Cells... |

| idMso | Control Type | Label |
|----------------------------------|---------------------|-----------------------|
| WindowsArrangeAll | button | Arrange All |
| WindowNew | button | New Window |
| ReviewAcceptOrRejectChangeDialog | button | Accept/Reject Changes |
| SymbolInsert | button | Symbol... |
| ReplaceDialog | button | Replace... |
| PagePrevious | button | Previous Page |
| PageNext | button | Next Page |
| TextBoxInsertVertical | toggleButton | Vertical Text Box |
| ObjectsRegroup | button | Regroup |
| PrintAreaSetPrintArea | button | Set Print Area |
| PasteFormatting | button | Paste Formatting |
| PasteValues | button | Paste Values |
| FillRight | button | Right |
| FillDown | button | Down |
| EqualSign | button | Equal Sign |
| PlusSign | button | Plus Sign |
| MinusSign | button | Minus Sign |
| MultiplicationSign | button | Multiplication Sign |
| DivisionSign | button | Division Sign |
| ExponentiationSign | button | Exponentiation Sign |
| ParenthesisLeft | button | Left Parenthesis |
| ParenthesisRight | button | Right Parenthesis |
| ColonSign | button | Colon |
| CommaSign | button | Comma |
| PercentSign | button | Percent Sign |
| DollarSign | button | Dollar Sign |
| FunctionWizard | button | Insert Function... |
| ConstrainNumeric | button | Constrain Numeric |
| LightShading | button | Light Shading |

| idMso | Control Type | Label |
|------------------------------------|---------------------|--------------------------|
| AccountingFormat | button | Accounting Number Format |
| PercentStyle | button | Percent Style |
| CommaStyle | button | Comma Style |
| DecimalsIncrease | button | Increase Decimal |
| DecimalsDecrease | button | Decrease Decimal |
| MergeCenter | toggleButton | Merge & Center |
| FontSizeIncrease | button | Increase Font Size |
| FontSizeDecrease | button | Decrease Font Size |
| TextOrientationVertical | toggleButton | Vertical Text |
| TextOrientationRotateUp | toggleButton | Rotate Text Up |
| TextOrientationRotateDown | toggleButton | Rotate Text Down |
| AlignDistributeHorizontallyClassic | button | Distribute Horizontally |
| ShapeScribble | toggleButton | Scribble |
| OutlineSymbolsShowHide | button | Show Outline Symbols |
| TableSelectVisibleCells | button | Select Visible Cells |
| SelectCurrentRegion | button | Select Current Region |
| FreezePanes | button | Freeze Panes |
| ZoomIn | button | Zoom In |
| ZoomOut | button | Zoom Out |
| FormControlRadioButton | button | Option Button |
| FormControlScrollBar | button | Scroll Bar |
| FormControlListBox | button | List Box |
| TraceDependentRemoveArrows | button | Remove Dependent Arrows |
| TraceDependents | button | Trace Dependents |
| TracePrecedentsRemoveArrows | button | Remove Precedent Arrows |
| TraceRemoveAllArrows | button | Remove Arrows |
| FileUpdate | button | Update File |
| ReadOnly | button | Toggle Read Only |
| AutoFilterClassic | button | AutoFilter |

| idMso | Control Type | Label |
|------------------------------------|---------------------|-----------------------------|
| Refresh | button | Refresh |
| PivotTableFieldSettings | button | Field Settings |
| PivotTableShowPages | button | Show Report Filter Pages... |
| OutlineShowDetail | button | Show Detail |
| TraceError | button | Trace Error |
| OutlineHideDetail | button | Hide Detail |
| AlignDistributeVerticallyClassic | button | Distribute Vertically |
| FormControlGroupBox | button | Group Box |
| FormControlSpinner | button | Spinner |
| TabOrder | button | Tab Order... |
| RunDialog | button | Run Dialog |
| FormControlCombinationListEdit | button | Combination List-Edit |
| FormControlCombinationDropDownEdit | button | Combination Drop-Down Edit |
| FormControlLabel | button | Label |
| TracePrecedents | button | Trace Precedents |
| CodeEdit | button | Code |
| PageBreakInsertOrRemove | button | Insert Page Break |
| QueryParameters | button | Parameters |
| RowHeight | button | Row Height... |
| ColumnWidth | button | Column Width... |
| OleObjectctInsert | button | Object... |
| SnapToGrid | toggleButton | Snap to Grid |
| ObjectsAlignLeft | button | Align Left |
| ObjectsAlignRight | button | Align Right |
| ObjectsAlignTop | button | Align Top |
| ObjectsAlignBottom | button | Align Bottom |
| ObjectsAlignCenterHorizontal | button | Align Center |
| ObjectsAlignMiddleVertical | button | Align Middle |
| ClipArtInsert | toggleButton | Clip Art... |

| idMso | Control Type | Label |
|----------------------------------|---------------------|--------------------------------------|
| ObjectRotateFree | button | Free Rotate |
| CombineCharacters | toggleButton | Yoko-Gumi |
| ViewNormalViewExcel | toggleButton | Normal |
| ViewPageBreakPreviewView | toggleButton | Page Break Preview |
| PictureCrop | toggleButton | Crop |
| FileCloseAll | button | Close All |
| FileSaveAs | button | Save As |
| AdvancedFileProperties | button | View Document Properties... |
| CopyAsPicture | button | Copy as Picture... |
| PasteSpecialDialog | button | Paste Special... |
| SelectAll | button | Select All |
| GoTo | button | Go To... |
| FileLinksToFiles | button | Edit Links to Files |
| HeaderFooterInsert | button | Header & Footer... |
| BulletsAndNumberingBulletsDialog | button | Bullets and Numbering... |
| AutoFormatDialog | button | AutoFormat... |
| MergeCells | button | Merge Cells |
| SplitCells | button | Split Cells... |
| ConvertTextToTable | button | Convert Text to Table... |
| ShowClipboard | button | Office Clipboard... |
| OutlookTaskCreate | button | Create Microsoft Office Outlook Task |
| WindowMinimize | button | Minimize |
| WindowRestore | button | Restore |
| WindowClose | button | Close |
| WindowSaveWorkspace | button | Save Workspace... |
| SheetDelete | button | Delete Sheet |
| SheetMoveOrCopy | button | Move or Copy Sheet... |
| ViewFormulaBar | checkBox | Formula Bar |
| SheetInsert | button | Insert Sheet |

| idMso | Control Type | Label |
|-------------------------|---------------------|--------------------------|
| FormatCellsDialog | button | Format Cells... |
| GoalSeek | button | Goal Seek... |
| ScenarioManager | button | Scenario Manager... |
| DataFormExcel | button | Form... |
| OutlineSubtotals | button | Subtotal |
| DataTable | button | Data Table... |
| Consolidate | button | Consolidate... |
| WindowHide | button | Hide |
| WindowUnhide | button | Unhide... |
| FillUp | button | Up |
| FillLeft | button | Left |
| FillAcrossWorksheets | button | Across Worksheets... |
| FillSeries | button | Series... |
| FillJustify | button | Justify |
| ClearFormats | button | Clear Formats |
| ClearContents | button | Clear Contents |
| ClearComments | button | Clear Comments |
| NamePasteName | button | Paste Names... |
| NameCreateFromSelection | button | Create from Selection... |
| NamesApply | button | Apply Names... |
| RowHeightAutoFit | button | AutoFit Row Height |
| RowsHide | button | Hide Rows |
| RowsUnhide | button | Unhide Rows |
| ColumnWidthAutoFit | button | AutoFit Column Width |
| ColumnsHide | button | Hide Columns |
| ColumnsUnhide | button | Unhide Columns |
| ColumnWidthDefault | button | Default Width... |
| SheetRename | button | Rename Sheet |
| SheetHide | button | Hide Sheet |

| idMso | Control Type | Label |
|------------------------------|---------------------|----------------------------|
| SheetUnhide | button | Unhide Sheet... |
| SheetProtect | button | Protect Sheet... |
| ReviewProtectWorkbook | toggleButton | Protect Workbook... |
| MacroRelativeReferences | toggleButton | Use Relative References |
| Filter | toggleButton | Filter |
| SortClear | button | Clear |
| AdvancedFilterDialog | button | Advanced... |
| OutlineAuto | button | Auto Outline |
| OutlineClear | button | Clear Outline |
| OutlineSettings | button | Group and Outline Settings |
| PrintPreviewClose | button | Close Print Preview |
| ZoomDialog | button | Zoom... |
| SortDialogClassic | button | Sort... |
| ExchangeFolder | button | Exchange Folder... |
| AddInManager | button | Add-Ins... |
| ViewCustomViews | button | Custom Views... |
| SheetBackground | button | Background... |
| ChartEditDataSource | button | Select Data... |
| ChartPlacement | button | Move Chart... |
| CalculateNow | button | Calculate Now |
| ObjectFormatDialog | button | Object... |
| Help | button | Help |
| PivotTableEnableSelection | toggleButton | Enable Selection |
| PivotTableListFormulas | button | List Formulas |
| PivotTableSelectData | button | Values |
| PivotTableSelectLabelAndData | button | Labels and Values |
| PivotTableSelectLabel | button | Labels |
| PasteAsPicture | button | Paste as Picture |
| PastePictureLink | button | Paste Picture Link |

| idMso | Control Type | Label |
|--|---------------------|------------------------|
| CalculateSheet | button | Calculate Sheet |
| TextOrientationAngleCounterclockwise | toggleButton | Angle Counterclockwise |
| TextOrientationAngleClockwise | toggleButton | Angle Clockwise |
| WebGoBack | button | Back |
| WebGoForward | button | Forward |
| SmartArtInsert | button | SmartArt... |
| ShapeRerouteConnectors | toggleButton | Reroute Connectors |
| ObjectNudgeUp | button | Up |
| ObjectNudgeDown | button | Down |
| ObjectNudgeLeft | button | Left |
| ObjectNudgeRight | button | Right |
| ShapeCurve | toggleButton | Curve |
| ShapeStraightConnector | toggleButton | Straight Connector |
| ShapeElbowConnector | toggleButton | Elbow Connector |
| ObjectFillMoreColorsDialog | button | More Fill Colors... |
| ObjectBorderOutlineColorMoreColorsDialog | button | More Outline Colors... |
| LineStyleDialog | button | More Lines... |
| ArrowsMore | button | More Arrows... |
| WordArtVerticalText | button | Vertical Text |
| ContrastMore | button | More Contrast |
| ContrastLess | button | Less Contrast |
| BrightnessMore | button | More Brightness |
| BrightnessLess | button | Less Brightness |
| ShadowNudgeUpClassic | button | Nudge Shadow Up |
| ShadowNudgeDownClassic | button | Nudge Shadow Down |
| ShadowNudgeLeftClassic | button | Nudge Shadow Left |
| ShadowNudgeRightClassic | button | Nudge Shadow Right |
| ObjectShadowColorMoreColorsDialog | button | More Shadow Colors... |
| _3DEffectColorPickerMoreClassic | button | More 3-D Colors... |

| idMso | Control Type | Label |
|--------------------------------|---------------------|-----------------------------|
| ShapeRectangle | toggleButton | Rectangle |
| ShapeRoundedRectangle | toggleButton | Rounded Rectangle |
| ShapeIsoscelesTriangle | toggleButton | Isosceles Triangle |
| ShapeOval | toggleButton | Oval |
| ShapeLeftBrace | toggleButton | Left Brace |
| ShapeRightBrace | toggleButton | Right Brace |
| ShapeArc | toggleButton | Arc |
| ShapeRightArrow | toggleButton | Right Arrow |
| ShapeDownArrow | toggleButton | Down Arrow |
| ShapeRoundedRectangularCallout | toggleButton | Rounded Rectangular Callout |
| ShapeStar | toggleButton | 5-Point Star |
| PictureReset | button | Reset Picture |
| _3DEffectsOnOffClassic | toggleButton | 3-D On/Off |
| _3DTiltDownClassic | button | Tilt Down |
| _3DTiltUpClassic | button | Tilt Up |
| _3DTiltLeftClassic | button | Tilt Left |
| _3DTiltRightClassic | button | Tilt Right |
| _3DSurfaceMaterialClassic | menu | Surface |
| _3DExtrusionPerspectiveClassic | toggleButton | Perspective |
| _3DExtrusionParallelClassic | toggleButton | Parallel |
| _3DLightingFlatClassic | toggleButton | Bright |
| _3DLightingNormalClassic | toggleButton | Normal |
| _3DLightingDimClassic | toggleButton | Dim |
| _3DSurfaceMatteClassic | toggleButton | Matte |
| _3DSurfacePlasticClassic | toggleButton | Plastic |
| _3DSurfaceMetalClassic | toggleButton | Metal |
| _3DSurfaceWireFrameClassic | toggleButton | Wire Frame |
| SnapToShapes | toggleButton | Snap to Shape |
| HyperlinkInsert | button | Hyperlink... |

| idMso | Control Type | Label |
|-----------------------------|---------------------|------------------------------|
| PrintAreaAddToPrintArea | button | Add to Print Area |
| PrintAreaClearPrintArea | button | Clear Print Area |
| PageBreaksResetAll | button | Reset All Page Breaks |
| ReviewNewComment | button | New Comment |
| ReviewPreviousComment | button | Previous |
| ReviewNextComment | button | Next |
| ReviewDeleteComment | button | Delete |
| ReviewShowOrHideComment | button | Show/Hide Comment |
| ReviewShowAllComments | toggleButton | Show All Comments |
| PivotTableFieldInsert | button | Calculated Field... |
| PivotTableCalculatedItem | button | Calculated Item... |
| PivotTableSelectEntireTable | button | Entire PivotTable |
| PivotTableOptions | button | Options |
| DesignMode | toggleButton | Design Mode |
| PhoneticGuideEdit | button | Edit Phonetic |
| PhoneticGuideSettings | button | Phonetic Settings... |
| PhoneticGuideFieldShow | toggleButton | Show Phonetic Field |
| CircularReferences | gallery | Circular References |
| VisualBasic | button | Visual Basic |
| BorderThickBottom | button | Thick Bottom Border |
| BorderTopAndBottom | button | Top and Bottom Border |
| BorderTopAndDoubleBottom | button | Top and Double Bottom Border |
| BorderTopAndThickBottom | button | Top and Thick Bottom Border |
| BordersAll | button | All Borders |
| BorderThickOutside | button | Thick Box Border |
| Font | comboBox | Font: |
| FontSize | comboBox | Font Size: |
| StyleGalleryClassic | gallery | Style: |
| ZoomClassic | gallery | Zoom: |

| idMso | Control Type | Label |
|------------------------------|---------------------|--------------------------|
| ScenarioGallery | gallery | Scenario: |
| DocumentLocation | comboBox | Address: |
| MergeCellsAcross | button | Merge Across |
| BorderInsideHorizontal | button | Inside Horizontal Border |
| BorderInsideVertical | button | Inside Vertical Border |
| BorderDiagonalDown | button | Diagonal Down Border |
| BorderDiagonalUp | button | Diagonal Up Border |
| TextDirectionLeftToRight | toggleButton | Left-to-Right |
| TextDirectionRightToLeft | toggleButton | Right-to-Left |
| ActiveXCheckBox | button | Check Box |
| FindDialogExcel | button | Find... |
| ActiveXTextBox | button | Text Box |
| ActiveXButton | button | Command Button |
| ActiveXRadioButton | button | Option Button |
| ActiveXListBox | button | List Box |
| ActiveXComboBox | button | Combo Box |
| ActiveXToggleButton | button | Toggle Button |
| ActiveXSpinButton | button | Spin Button |
| ActiveXScrollBar | button | Scroll Bar |
| ActiveXLabel | button | Label |
| ShadowSemitransparentClassic | toggleButton | Semitransparent Shadow |
| RightToLeftDocument | toggleButton | Right-to-Left Document |
| EditQuery | button | Edit Query... |
| DataRangeProperties | button | Properties |
| RefreshAll | button | Refresh All |
| RefreshCancel | button | Cancel Refresh |
| RefreshStatus | button | Refresh Status |
| PasteLink | button | Paste Link... |
| ClearAll | button | Clear All |

| idMso | Control Type | Label |
|--------------------------------------|---------------------|----------------------------------|
| DataValidation | button | Data Validation... |
| DataValidationCircleInvalid | button | Circle Invalid Data |
| ReviewShareWorkbook | button | Share Workbook... |
| ReviewHighlightChanges | button | Highlight Changes... |
| CompareAndMergeWorkbooks | button | Compare and Merge Workbooks... |
| DatabaseQueryNew | button | New Database Query... |
| DataValidationClearValidationCircles | button | Clear Validation Circles |
| ActiveXImage | button | Image |
| ShadowOnOrOffClassic | toggleButton | Shadow On/Off |
| ObjectSetShapeDefaults | button | Set AutoShape Defaults |
| FileSendAsAttachment | button | E-mail |
| FileNewDefault | button | New |
| FilePrintQuick | button | Quick Print |
| PictureInsertFromFile | button | Picture... |
| ShapeStraightConnectorArrow | toggleButton | Straight Arrow Connector |
| ShapeElbowConnectorArrow | toggleButton | Elbow Arrow Connector |
| PasteAsHyperlink | button | Paste as Hyperlink |
| ParagraphDistributed | toggleButton | Distributed |
| PictureSetTransparentColor | toggleButton | Set Transparent Color |
| PivotTableSolveOrder | button | Solve Order... |
| PivotTableReport | button | PivotTable and PivotChart Wizard |
| ReviewProtectAndShareWorkbook | button | Protect Sharing |
| OutlineGroup | button | Group... |
| OutlineUngroup | button | Ungroup... |
| IndentIncreaseExcel | button | Increase Indent |
| IndentDecreaseExcel | button | Decrease Indent |
| HangulHanjaConversion | button | Hangul Hanja Conversion... |
| MacroSecurity | button | Macro Security |
| WebPagePreview | button | Web Page Preview |

| idMso | Control Type | Label |
|-----------------------------|---------------------|-------------------------|
| SendCopySendNow | button | Send Now |
| SendCopySelectNames | button | Select Names |
| SendCopySendToMailRecipient | toggleButton | Mail Recipient |
| ComAddInsDialog | button | COM Add-Ins... |
| PivotChartInsertClassic | button | PivotChart |
| PivotFieldListShowHide | toggleButton | Field List |
| FileSaveAsWebPage | button | Save as Web Page... |
| GetExternalDataFromWeb | button | From Web |
| WebOptionsDialog | button | Web Options... |
| GetExternalDataFromText | button | From Text |
| FilePublishAsWebPage | button | Publish as Web Page... |
| PivotTableOlapOffline | button | Offline OLAP... |
| ObjectsMultiSelect | button | Select Multiple Objects |
| TextDirectionContext | toggleButton | Context |
| CalculateFull | button | TBA |
| WatchWindow | toggleButton | Watch Window |
| FormulaEvaluate | button | Evaluate Formula |
| AutoSumAverage | button | Average |
| AutoSumCount | button | Count Numbers |
| AutoSumMax | button | Max |
| AutoSumMin | button | Min |
| AutoSumMoreFunctions | button | More Functions... |
| PasteFormulas | button | Formulas |
| PasteNoBorders | button | No Borders |
| PasteTranspose | button | Transpose |
| DrawingCanvasFit | button | Fit |
| DrawingCanvasResize | button | Resize |
| DrawingCanvasExpand | button | Expand |
| ReviewSendForReview | button | Send for Review... |

| idMso | Control Type | Label |
|--------------------------------|---------------------|-------------------------------|
| BorderDrawMenu | splitButton | Draw Border |
| BorderErase | toggleButton | Erase Border |
| BorderStyle | dropDown | Line Style |
| ShowFormulas | toggleButton | Show Formulas |
| PivotTableOlapPropertyFields | button | Property Fields... |
| TranslationPane | button | Translate... |
| ErrorChecking | button | Error Checking... |
| FileCheckOut | button | Check Out |
| FileCheckIn | button | Check In |
| PivotTableGenerateGetPivotData | checkBox | Generate GetPivotData |
| ReviewReplyWithChanges | button | Reply with Changes... |
| ReviewEndReview | button | End Review... |
| BorderDrawLine | toggleButton | Draw Border |
| BorderDrawGrid | toggleButton | Draw Border Grid |
| GetExternalDataImportClassic | button | Import External Data |
| PicturesCompress | button | Compress Pictures... |
| VerticallyDistributed | button | Vertically Distributed |
| ReviewAllowUsersToEditRanges | button | Allow Users to Edit Ranges... |
| SpeakCells | button | Speak Cells |
| SpeakStop | button | Stop Speaking |
| SpeakByRows | toggleButton | By Rows |
| SpeakByColumns | toggleButton | By Columns |
| SpeakOnEnter | toggleButton | On Enter |
| TableInsertExcel | button | Table |
| ResearchPane | toggleButton | Research... |
| TableStyleTotalsRow | checkBox | Total Row |
| TableRowsInsertAboveExcel | button | Insert Table Rows Above |
| TableRowsDeleteExcel | button | Delete Table Rows |
| TableConvertToRange | button | Convert to Range |

| idMso | Control Type | Label |
|--------------------------------------|---------------------|------------------------------------|
| PrintListRange | button | Print List |
| FileInternetFax | button | Internet Fax |
| XmlExport | button | Export |
| XmlImport | button | Import |
| ViewDocumentActionsPane | toggleButton | Document Actions |
| ReviewShowInk | toggleButton | Show Ink |
| TableColumnsInsertLeftExcel | button | Insert Table Columns to the Left |
| TableColumnsDeleteExcel | button | Delete Table Columns |
| InkDeleteAllInk | button | Delete All Ink |
| TableUnlinkExternalData | button | Unlink |
| TableExportTableToSharePointList | button | Export Table to SharePoint List... |
| PrivacyOptionsDialog | button | Privacy Options... |
| ListSynchronize | button | Synchronize List |
| ChangesDiscardAndRefresh | button | Discard Changes and Refresh |
| TableOpenInBrowser | button | Open in Browser |
| TableResize | button | Resize Table |
| XmlExpansionPacksExcel | button | Expansion Packs |
| FileVersionHistory | button | View Version History |
| XmlDataRefresh | button | Refresh Data |
| XmlMapProperties | button | Map Properties |
| WindowSideBySideSynchronousScrolling | toggleButton | Synchronous Scrolling |
| WindowResetPosition | button | Reset Window Position |
| InkColorMoreColorsDialog | button | More Ink Colors... |
| ContactUs | button | Contact Us... |
| FilePermissionUnrestricted | toggleButton | Unrestricted Access |
| FilePermissionDoNotDistribute | toggleButton | Restricted Access |
| FilePermissionView | button | View Permission |
| FilePermission | button | Permission |
| Thesaurus | button | Thesaurus... |

| idMso | Control Type | Label |
|---------------------------|---------------------|---------------------|
| InkingStart | button | Start Inking |
| CheckForUpdates | button | Check for Updates |
| InkCopyAsText | button | Copy Ink As Text |
| InkEraser | toggleButton | Eraser |
| FilePermissionRestrictAs | button | Manage Credentials |
| Connections | button | Connections |
| FileViewDigitalSignatures | toggleButton | View Signatures |
| FileWorkflowTasks | button | View Workflow Tasks |
| FileStartWorkflow | button | Workflows |
| SignatureLineInsert | button | Signature Line |
| LabelInsert | button | Label |
| BarcodeInsert | button | Barcode |
| ViewPageLayoutView | toggleButton | Page Layout |
| PivotClearAll | button | Clear All |
| ChartStylesGallery | gallery | Quick Styles |
| ChartLayoutGallery | gallery | Quick Layout |
| ChartSaveTemplates | button | Save As Template |
| ChartAxisTitles | menu | Axis Titles |
| ChartAxes | menu | Axes |
| ChartGridlines | menu | Gridlines |
| ChartFormatSelection | button | Format Selection |
| ChartElementSelector | comboBox | Chart Elements |
| PageMarginsGallery | gallery | Margins |
| TabPictureToolsFormat | tab | Format |
| TabDrawingToolsFormat | tab | Format |
| ShapesInsertGallery | gallery | Shapes |
| ShapeChangeShapeGallery | gallery | Change Shape |
| ShapeFillTextureGallery | gallery | Texture |
| ShapeStylesGallery | gallery | Quick Styles |

| idMso | Control Type | Label |
|--------------------------|---------------------|---------------|
| PageOrientationGallery | gallery | Orientation |
| FileServerTasksMenu | menu | Server |
| FileSendMenu | menu | Send |
| TabInsert | tab | Insert |
| TabPageLayoutExcel | tab | Page Layout |
| TabView | tab | View |
| GroupFont | group | Font |
| GroupStyles | group | Styles |
| GroupProofing | group | Proofing |
| GroupInsertIllustrations | group | Illustrations |
| GroupShapes | group | Insert Shapes |
| GroupPageSetup | group | Page Setup |
| GroupComments | group | Comments |
| GroupPictureSize | group | Size |
| GroupDrawBorders | group | Draw Borders |
| GroupTableProperties | group | Properties |
| GroupTableTools | group | Tools |
| GroupArrange | group | Arrange |
| GroupShapeStyles | group | Shape Styles |
| TabFormulas | tab | Formulas |
| TabData | tab | Data |
| TabReview | tab | Review |
| GroupChartLayouts | group | Chart Layouts |
| GroupChartStyles | group | Chart Styles |
| GroupChartAxes | group | Axes |
| GroupChartShapes | group | Insert |
| GroupNumber | group | Number |
| GroupAlignmentExcel | group | Alignment |
| GroupCells | group | Cells |

| idMso | Control Type | Label |
|-----------------------------|---------------------|-------------------|
| GroupSortFilter | group | Sort & Filter |
| GroupInsertTablesExcel | group | Tables |
| GroupPageLayoutScaleToFit | group | Scale to Fit |
| GroupPageLayoutSheetOptions | group | Sheet Options |
| GroupFunctionLibrary | group | Function Library |
| GroupNamedCells | group | Defined Names |
| GroupFormulaAuditing | group | Formula Auditing |
| GroupGetExternalData | group | Get External Data |
| GroupConnections | group | Connections |
| GroupOutline | group | Outline |
| GroupDataTools | group | Data Tools |
| GroupChangesExcel | group | Changes |
| ObjectEffectPresetGallery | gallery | Preset |
| PictureEffectsPresetGallery | gallery | Preset |
| _3DRotationGallery | gallery | 3-D Rotation |
| TabSmartArtToolsDesign | tab | Design |
| TabSmartArtToolsFormat | tab | Format |
| TabChartToolsDesign | tab | Design |
| TabChartToolsLayout | tab | Layout |
| TabChartToolsFormat | tab | Format |
| ShapeFillColorPicker | gallery | Shape Fill |
| OutlineColorPicker | gallery | Picture Border |
| FileDocumentInspect | button | Inspect Document |
| AlignLeftToRightMenu | splitButton | Left-to-Right |
| GroupControls | group | Controls |
| GroupZoom | group | Zoom |
| ArrowStyleGallery | gallery | Arrows |
| OutlineDashesGallery | gallery | Dashes |
| OutlineWeightGallery | gallery | Weight |

| idMso | Control Type | Label |
|-----------------------------|---------------------|--------------------------|
| GroupPictureTools | group | Adjust |
| GroupSize | group | Size |
| FormatCellsNumberDialog | button | Format Cell Number |
| FormatCellsFontDialog | button | Format Cell Font |
| CellAlignmentOptions | button | Format Cell Alignment |
| PageSetupPageDialog | button | Page Setup |
| PageSetupSheetDialog | button | Sheet Options |
| TabPivotTableToolsOptions | tab | Options |
| PivotTableLayoutGrandTotals | menu | Grand Totals |
| TabPivotTableToolsDesign | tab | Design |
| GroupPivotTableActiveField | group | Active Field |
| GroupPivotTableLayout | group | Layout |
| GroupPivotTableSort | group | Sort |
| GroupPivotTableShowHide | group | Show/Hide |
| PivotTableLayoutSubtotals | menu | Subtotals |
| GroupPivotTableGroup | group | Group |
| GroupPivotTableTools | group | Tools |
| GroupPivotTableData | group | Data |
| GroupPivotTableOptions | group | PivotTable |
| GroupPivotTableStyles | group | PivotTable Styles |
| GroupPivotTableStyleOptions | group | PivotTable Style Options |
| WrapText | toggleButton | Wrap Text |
| ClearMenu | menu | Clear |
| ReviewTrackChangesMenu | menu | Track Changes |
| ObjectAlignMenu | menu | Align |
| ObjectRotateGallery | gallery | Rotate |
| FillMenu | menu | Fill |
| OrientationMenu | menu | Orientation |
| MergeCenterMenu | splitButton | Merge |

| idMso | Control Type | Label |
|---------------------------------|---------------------|-------------------------------|
| AutoSumMenu | splitButton | AutoSum |
| PrintAreaMenu | menu | Print Area |
| PageBreakMenu | menu | Breaks |
| NameDefineMenu | splitButton | Define Name |
| RefreshMenu | splitButton | Refresh |
| WhatIfAnalysisMenu | menu | What-If Analysis |
| PivotTableFormulasMenu | menu | Formulas |
| PivotTableOlapTools | menu | OLAP tools |
| PivotTableOptionsMenu | splitButton | Table Options |
| ErrorCheckingMenu | splitButton | Error Checking |
| TraceRemoveArrowsMenu | splitButton | Remove Arrows |
| SortFilterMenu | menu | Sort & Filter |
| FontColorPicker | gallery | Font Color |
| CellFillColorPicker | gallery | Shading |
| BorderDoubleBottom | button | Bottom Double Border |
| TabHome | tab | Home |
| ChartTitle | gallery | Chart Title |
| ChartPrimaryHorizontalAxisTitle | gallery | Primary Horizontal Axis Title |
| ChartPrimaryVerticalAxisTitle | gallery | Primary Vertical Axis Title |
| ChartDepthAxisTitle | gallery | Depth Axis Title |
| ChartLegend | gallery | Legend |
| ChartDataLabel | gallery | Data Labels |
| ChartPrimaryHorizontalGridlines | gallery | Primary Horizontal Gridlines |
| ChartPrimaryVerticalGridlines | gallery | Primary Vertical Gridlines |
| ChartDepthGridlines | gallery | Depth Gridlines |
| ChartPrimaryHorizontalAxis | gallery | Primary Horizontal Axis |
| ChartPrimaryVerticalAxis | gallery | Primary Vertical Axis |
| ChartDepthAxis | gallery | Depth Axis |
| ChartDataTable | gallery | Data Table |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|--------------------------|
| ChartTrendline | gallery | Trendline |
| ChartErrorBars | gallery | Error Bars |
| ChartLines | gallery | Lines |
| ChartUpDownBars | gallery | Up/Down Bars |
| ChartPlotArea | gallery | Plot Area |
| ChartWall | gallery | Chart Wall |
| ChartFloor | gallery | Chart Floor |
| SmartArtAddShape | button | Add Shape |
| SmartArtLargerShape | button | Larger |
| SmartArtSmallerShape | button | Smaller |
| SmartArtResetGraphic | button | Reset Graphic |
| SmartArtTextPane | toggleButton | Text Pane |
| SmartArtEditIn2D | toggleButton | Edit in 2-D |
| SmartArtLayoutGallery | gallery | Change Layout |
| SmartArtMoreLayoutsDialog | button | More Layouts... |
| SmartArtStylesGallery | gallery | Quick Styles |
| SmartArtChangeColorsGallery | gallery | Change Colors |
| ObjectEffectSoftEdgesGallery | gallery | Soft Edges |
| ObjectEffectGlowGallery | gallery | Glow |
| GradientGallery | gallery | Gradient |
| ObjectEffectShadowGallery | gallery | Shadow |
| WordArtInsertGallery | gallery | WordArt |
| TextEffectTransformGallery | gallery | Transform |
| TabHeaderAndFooterToolsDesign | tab | Design |
| HeaderFooterHeaderGallery | gallery | Header |
| HeaderFooterFooterGallery | gallery | Footer |
| GroupHeaderFooterElements | group | Header & Footer Elements |
| HeaderFooterPageNumberInsertExcel | button | Page Number |
| HeaderFooterNumberOfPagesInsert | button | Number of Pages |

| idMso | Control Type | Label |
|--|--------------|----------------------------------|
| HeaderFooterCurrentDate | button | Current Date |
| HeaderFooterCurrentTimeInsert | button | Current Time |
| HeaderFooterFilePathInsert | button | File Path |
| HeaderFooterFileNameInsert | button | File Name |
| HeaderFooterSheetNameInsert | button | Sheet Name |
| HeaderFooterAlignMargins | checkBox | Align with Page Margins |
| HeaderFooterPictureInsert | button | Picture |
| HeaderFooterFormatPicture | button | Format Picture |
| GroupHeaderFooterOptions | group | Options |
| HeaderFooterDifferentOddEvenPageExcel | checkBox | Different Odd & Even Pages |
| HeaderFooterDifferentFirstPageExcel | checkBox | Different First Page |
| HeaderFooterScaleWithDocument | checkBox | Scale with Document |
| SheetTabColorGallery | gallery | Tab Color |
| FontShadingColorMoreColorsDialog | button | More Colors... |
| FontColorMoreColorsDialogExcel | button | More Colors... |
| BorderMoreColorsDialog | button | More Colors... |
| SheetTabColorMoreColorsDialog | button | More Colors... |
| PivotTableNewStyle | button | New PivotTable Style... |
| PivotPlusMinusFieldHeadersShowHide | toggleButton | Field Headers |
| PivotTableExpandField | button | Expand Entire Field |
| PivotCollapseField | button | Collapse Entire Field |
| ConditionalFormattingDataBarsGallery | gallery | Data Bars |
| ConditionalFormattingColorScalesGallery | gallery | Color Scales |
| ConditionalFormattingIconSetsGallery | gallery | Icon Sets |
| ConditionalFormattingDataBarsMoreOptions | button | More Rules... |
| ConditionalFormattingColorScalesMore | button | More Rules... |
| ConditionalFormattingIconSetsMore | button | More Rules... |
| TableColumnsInsertRightExcel | button | Insert Table Column to the Right |
| TableRowsInsertBelowExcel | button | Insert Table Row Below |

| idMso | Control Type | Label |
|---|--------------|----------------------------------|
| ConditionalFormattingHighlightBetween | button | Between... |
| ConditionalFormattingClearSelectedCells | button | Clear Rules from Selected Cells |
| ConditionalFormattingClearSheet | button | Clear Rules from Entire Sheet |
| ConditionalFormattingClearTable | button | Clear Rules from This Table |
| ConditionalFormattingClearPivotTable | button | Clear Rules from This PivotTable |
| PivotTableStylesGallery | gallery | Quick Styles |
| FormatAsTableGallery | gallery | Format as Table |
| TableStylesGalleryExcel | gallery | Quick Styles |
| ConditionalFormattingsManage | button | Manage Rules... |
| ConditionalFormattingHighlightGreaterThan | button | Greater Than... |
| ConditionalFormattingHighlightLessThan | button | Less Than... |
| ConditionalFormattingHighlightEqualTo | button | Equal To... |
| ConditionalFormattingHighlightTextContaining | button | Text that Contains... |
| ConditionalFormattingHighlightDateOccuring | button | A Date Occurring... |
| ConditionalFormattingHighlightDuplicateValues | button | Duplicate Values... |
| ConditionalFormattingTopNItems | button | Top 10 Items... |
| ConditionalFormattingTopNPercent | button | Top 10 %... |
| ConditionalFormattingBottomNItems | button | Bottom 10 Items... |
| ConditionalFormattingBottomNPercent | button | Bottom 10 %... |
| ConditionalFormattingAboveAverage | button | Above Average... |
| ConditionalFormattingBelowAverage | button | Below Average... |
| RemoveDuplicates | button | Remove Duplicates |
| FilterReapply | button | Reapply |
| ThemeColorsGallery | gallery | Colors |
| PivotTableInsert | button | PivotTable |
| PivotChartInsert | button | PivotChart |
| PivotTableMove | button | Move PivotTable |
| PivotTableChangeDataSource | button | Change Data Source... |
| GroupSmartArtLayouts | group | Layouts |

| idMso | Control Type | Label |
|---|--------------|---|
| GroupSmartArtQuickStyles | group | SmartArt Styles |
| GroupSmartArtCreateGraphic | group | Create Graphic |
| GroupSmartArtReset | group | Reset |
| GroupSmartArtSize | group | Size |
| ConditionalFormattingHighlightCellsMenu | menu | Highlight Cells Rules |
| ConditionalFormattingTopBottomMenu | menu | Top/Bottom Rules |
| FormatCellsMenu | menu | Format |
| ConditionalFormattingClearMenu | menu | Clear Rules |
| ThemeSearchOfficeOnline | button | More Themes on Microsoft Office Online... |
| FontColorCycle | button | Color |
| TabAddIns | tab | Add-Ins |
| CellsInsertSmart | button | Insert Cells |
| CellsDeleteSmart | button | Delete Cells... |
| ObjectEditShapeMenu | menu | Edit Shape |
| PivotTableLayoutReportLayout | menu | Report Layout |
| PivotTableLayoutShowInCompactForm | button | Show in Compact Form |
| PivotTableLayoutShowInOutlineForm | button | Show in Outline Form |
| PivotTableLayoutShowInTabularForm | button | Show in Tabular Form |
| PivotTableClearMenu | menu | Clear |
| ConditionalFormattingTopBottomMore | button | More Rules... |
| ConditionalFormattingHighlightRulesMore | button | More Rules... |
| CellStylesGallery | gallery | Cell Styles |
| CellStyleNew | button | New Cell Style... |
| CellStylesMerge | button | Merge Styles... |
| TableStyleNew | button | New Table Style... |
| TableStyleClear | button | Clear |
| TableStyleHeaderRow | checkBox | Header Row |
| FilePublishExcelServices | button | Excel Services |
| PivotTableOlapConvertToFormulas | button | Convert to Formulas |

| idMso | Control Type | Label |
|--|--------------|---------------------------|
| PivotTableLayoutBlankRows | menu | Blank Rows |
| TableStyleFirstColumn | checkBox | First Column |
| TableStyleLastColumn | checkBox | Last Column |
| TableStyleBandedRows | checkBox | Banded Rows |
| TableStyleBandedColumns | checkBox | Banded Columns |
| TableStyleRowHeaders | checkBox | Row Headers |
| TableStyleColumnHeaders | checkBox | Column Headers |
| TableSummarizeWithPivot | button | Summarize with PivotTable |
| ConnectionProperties | button | Connection Properties... |
| PivotClearFilters | button | Clear Filters |
| GetExternalDataFromAccess | button | From Access |
| GetExternalDataFromOtherSources | gallery | From Other Sources |
| GetExternalDataExistingConnections | button | Existing Connections |
| GroupThemesExcel | group | Themes |
| PivotPlusMinusButtonsShowHide | toggleButton | +/- Buttons |
| FileSaveAsPdfOrXps | button | Publish as PDF or XPS |
| FileSaveAsExcelOpenDocumentSpreadsheet | button | OpenDocument Spreadsheet |
| MoreControlsDialog | button | More Controls... |
| GroupCode | group | Code |
| TabDeveloper | tab | Developer |
| GroupXml | group | XML |
| PageScaleToFitWidth | comboBox | Width: |
| PageScaleToFitHeight | comboBox | Height: |
| SelectMenuExcel | menu | Find & Select |
| GoToSpecial | button | Go To Special... |
| GoToFormulas | button | Formulas |
| GoToComments | button | Comments |
| GoToConditionalFormatting | button | Conditional Formatting |
| GoToConstants | button | Constants |

| idMso | Control Type | Label |
|---|--------------|----------------------------------|
| GoToDataValidation | button | Data Validation |
| PrintTitles | button | Print Titles |
| NameUseInFormula | gallery | Use in Formula |
| CalculationOptionsMenu | menu | Calculation Options |
| CalculationOptionsManually | toggleButton | Manual |
| CalculationOptionsAutomatically | toggleButton | Automatic |
| CalculationOptionsAutomaticallyExceptDataTables | toggleButton | Automatic Except for Data Tables |
| XmlSource | toggleButton | Source... |
| GroupClipboard | group | Clipboard |
| GroupInsertLinks | group | Links |
| GroupInsertBarcode | group | Barcode |
| GroupCalculation | group | Calculation |
| BordersGallery | splitButton | Borders |
| BordersMoreDialog | button | More Borders... |
| PageScaleToFitOptionsDialog | button | More Pages... |
| PageBreakInsertExcel | button | Insert Page Break |
| PageBreakRemove | button | Remove Page Break |
| PageSizeGallery | gallery | Size |
| ObjectPictureFill | button | Picture... |
| PivotTableGroupSelection | button | Group Selection |
| PivotTableGroupField | button | Group Field |
| WindowSwitchWindowsMenuExcel | menu | Switch Windows |
| ThemeColorsCreateNew | button | Create New Theme Colors... |
| ThemeFontsCreateNew | button | Create New Theme Fonts... |
| ShapeFillMoreGradientsDialog | button | More Gradients... |
| ShadowOptionsDialog | button | Shadow Options... |
| MarginsCustomMargins | button | Custom Margins... |
| FunctionsRecentlyUsedtInsertGallery | gallery | Recently Used |
| FunctionsFinancialInsertGallery | gallery | Financial |

| idMso | Control Type | Label |
|---------------------------------------|---------------------|-----------------------------------|
| FunctionsDateTimeInsertGallery | gallery | Date & Time |
| FunctionsMathTrigInsertGallery | gallery | Math & Trig |
| FunctionsTextInsertGallery | gallery | Text |
| FunctionsLogicalInsertGallery | gallery | Logical |
| FunctionsStatisticalInsertGallery | gallery | Statistical |
| FunctionsLookupReferenceInsertGallery | gallery | Lookup & Reference |
| FunctionsInformationInsertGallery | gallery | Information |
| TabTableToolsDesignExcel | tab | Design |
| GroupTableStyleOptions | group | Table Style Options |
| GroupTableExternalData | group | External Table Data |
| GroupEditingExcel | group | Editing |
| FileCompatibilityChecker | button | Run Compatibility Checker |
| ThemeFontsGallery | gallery | Fonts |
| ThemeEffectsGallery | gallery | Effects |
| FileProperties | toggleButton | Properties |
| TabPrintPreview | tab | Print Preview |
| GroupPrintPreviewPrint | group | Print |
| GroupPrintPreviewPreview | group | Preview |
| SortDialog | button | Sort... |
| SortAscendingExcel | button | Sort Ascending |
| SortDescendingExcel | button | Sort Descending |
| SortCustomExcel | button | Custom Sort... |
| FileCreateDocumentWorkspace | toggleButton | Create Document Workspace |
| FileSaveToDocumentManagementServer | button | Document Management Server |
| FileDocumentManagementInformation | toggleButton | Document Management Information |
| QuickAccessToolbarCustomization | button | Customize Quick Access Toolbar... |
| FilePrepareMenu | menu | Prepare |
| FileMarkAsFinal | toggleButton | Mark as Final |
| FileAddDigitalSignature | button | Add a Digital Signature |

| idMso | Control Type | Label |
|---------------------------------------|---------------------|---------------------------|
| SignatureServicesAdd | button | Add Signature Services... |
| TextBoxInsertMenu | splitButton | Text Box |
| TextBoxInsertHorizontal | toggleButton | Horizontal Text Box |
| ControlProperties | button | Properties |
| ViewCode | button | View Code |
| GroupHeaderFooterNavigation | group | Navigation |
| ShadowStyleGalleryClassic | gallery | Shadow Effects |
| GroupInkSelect | group | Select |
| NumberFormatGallery | comboBox | Number Format |
| NumberFormatsDialog | button | More Number Formats... |
| PageSizeMorePaperSizesDialogExcel | button | More Paper Sizes... |
| PictureBrightnessGallery | gallery | Brightness |
| PictureContrastGallery | gallery | Contrast |
| SmartArtAddShapeAfter | button | Add Shape After |
| SmartArtAddShapeBefore | button | Add Shape Before |
| SmartArtAddShapeAbove | button | Add Shape Above |
| SmartArtAddShapeBelow | button | Add Shape Below |
| SmartArtAddAssistant | button | Add Assistant |
| ChartSwitchRowColumn | button | Switch Row/Column |
| ChartChangeType | button | Change Chart Type... |
| GroupChartData | group | Data |
| GroupChartLocation | group | Location |
| GroupChartType | group | Type |
| _3DRotationOptionsDialog | button | 3-D Rotation Options... |
| _3DBevelOptionsDialog | button | 3-D Options... |
| SelectionPane | toggleButton | Selection Pane... |
| SmartArtOrganizationChartLeftHanging | button | Left Hanging |
| SmartArtOrganizationChartRightHanging | button | Right Hanging |
| SmartArtOrganizationChartBoth | button | Both |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|----------------------------|
| SmartArtOrganizationChartStandard | button | Standard |
| SmartArtRightToLeft | toggleButton | Right to Left |
| ViewMessageBar | checkBox | Message Bar |
| NameManager | button | Name Manager |
| NameDefine | button | Define Name... |
| AccountingFormatExcel | button | |
| AccountingFormatMoreExcel | button | More Accounting Formats... |
| ShapeStylesOtherThemeFillsGallery | gallery | Other Theme Fills |
| SmartArtOrganizationChartMenu | menu | Layout |
| _3DEffectsGalleryClassic | gallery | 3-D Effects |
| _3DDirectionGalleryClassic | gallery | Direction |
| _3DLightingGalleryClassic | gallery | Lighting |
| GroupShadowEffects | group | Shadow Effects |
| Group3DEffects | group | 3-D Effects |
| ShadowColorPickerClassic | gallery | Shadow Color |
| _3DEffectColorPickerClassic | gallery | 3-D Color |
| ControlsGallery | gallery | Insert |
| GoToHeader | button | Go to Header |
| GoToFooter | button | Go to Footer |
| UnderlineGallery | splitButton | Underline |
| GroupAddInsMenuCommands | group | Menu Commands |
| GroupAddInsToolbarCommands | group | Toolbar Commands |
| ViewFreezePanelsGallery | gallery | Freeze Panes |
| GroupInk | group | Ink |
| TabInkToolsPens | tab | Pens |
| GroupInkPens | group | Pens |
| GroupInkClose | group | Close |
| InkBallpointPen | toggleButton | Ballpoint Pen |
| InkFeltTipPen | toggleButton | Felt Tip Pen |

| idMso | Control Type | Label |
|--|--------------|---|
| InkHighlighter | toggleButton | Highlighter |
| PhoneticGuideMenu | splitButton | Phonetic Guide |
| MarginsShowHide | checkBox | Show Margins |
| _3DSurfaceMaterialGalleryClassic | gallery | Surface |
| _3DExtrusionDepthGalleryClassic | gallery | Depth |
| GroupHeaderFooter | group | Header & Footer |
| FunctionsCubeInsertGallery | gallery | Cube |
| FunctionsEngineeringInsertGallery | gallery | Engineering |
| ThemeSaveCurrent | button | Save Current Theme... |
| ThemesGallery | gallery | Themes |
| ChartResetToMatchStyle | button | Reset to Match Style |
| Chart3DView | button | 3-D Rotation... |
| ObjectSizeAndPropertiesDialog | button | Size and Properties... |
| ShapeConvertToFreeform | button | Convert to Freeform |
| ReflectionGallery | gallery | Reflection |
| PictureRecolorGallery | gallery | Recolor |
| SmartArtPromote | button | Promote |
| SmartArtDemote | button | Demote |
| TabPivotChartToolsAnalyze | tab | Analyze |
| GroupPivotChartShowOrHide | group | Show/Hide |
| GroupPivotChartData | group | Data |
| GroupPivotChartActiveField | group | Active Field |
| PivotChartFilterShow | toggleButton | PivotChart Filter |
| ChartTitleOptionsDialog | button | More Title Options... |
| ChartLegendOptionsDialogDialog | button | More Legend Options... |
| ChartDataLabelDialog | button | More Data Label Options... |
| ChartPrimaryHorizontalAxisTitleOptionsDialog | button | More Primary Horizontal Axis Title Options... |
| ChartPrimaryVerticalAxisTitleOptionsDialog | button | More Primary Vertical Axis Title Options... |

| idMso | Control Type | Label |
|--|--------------|---|
| ChartSecondaryHorizontalAxisTitleOptionsDialog | button | More Secondary Horizontal Axis Title Options... |
| ChartSecondaryVerticalAxisTitleOptionsDialog | button | More Secondary Vertical Axis Title Options... |
| ChartDepthAxisTitleOptionsDialog | button | More Depth Axis Title Options... |
| ChartPrimaryHorizontalGridlinesOptionsDialog | button | More Primary Horizontal Gridlines Options... |
| ChartPrimaryVerticalGridlinesOptionsDialog | button | More Primary Vertical Gridlines Options... |
| ChartSecondaryHorizontalGridlinesOptionsDialog | button | More Secondary Horizontal Gridlines Options... |
| ChartSecondaryVerticalGridlinesOptionsDialog | button | More Secondary Vertical Gridlines Options... |
| ChartDepthGridlinesOptionsDialog | button | More Depth Gridlines Options... |
| ChartPrimaryHorizontalAxisOptionsDialog | button | More Primary Horizontal Axis Options... |
| ChartPrimaryVerticalAxisOptionsDialog | button | More Primary Vertical Axis Options... |
| ChartSecondaryHorizontalAxisOption | button | More Secondary Horizontal Axis Options... |
| ChartSecondaryVerticalAxisOptionsDialog | button | More Secondary Vertical Axis Options... |
| ChartDepthAxisOptionsDialog | button | More Depth Axis Options... |
| ChartDataTableOptionsDialog | button | More Data Table Options... |
| ChartTrendlineOptionsDialog | button | More Trendline Options... |
| ChartErrorBarsOptionsDialog | button | More Error Bars Options... |
| ChartUpDownBarsOptionsDialog | button | More Up/Down Bars Options... |
| ChartPlotAreaOptionsDialog | button | More Plot Area Options... |
| ChartWallOptionsDialog | button | More Walls Options... |
| ChartFloorOptionsDialog | button | More Floor Options... |
| ChartSecondaryHorizontalAxisTitle | gallery | Secondary Horizontal Axis Title |
| ChartSecondaryVerticalAxisTitle | gallery | Secondary Vertical Axis Title |
| ChartSecondaryHorizontalGridlines | gallery | Secondary Horizontal Gridlines |
| ChartSecondaryVerticalGridlines | gallery | Secondary Vertical Gridlines |
| ChartSecondaryHorizontalAxis | gallery | Secondary Horizontal Axis |

| idMso | Control Type | Label |
|----------------------------|---------------------|-------------------------|
| ChartSecondaryVerticalAxis | gallery | Secondary Vertical Axis |
| GroupAddInsCustomToolbars | group | Custom Toolbars |
| ConditionalFormattingMenu | menu | Conditional Formatting |
| TabPivotChartToolsDesign | tab | Design |
| TabPivotChartToolsLayout | tab | Layout |
| TabPivotChartToolsFormat | tab | Format |
| ObjectBringToFrontMenu | splitButton | Bring to Front |
| ObjectSendToBackMenu | splitButton | Send to Back |
| ObjectsGroupMenu | menu | Group |
| SignatureLineInsertMenu | splitButton | Signature Line |
| FileSaveAsExcel97_2003 | button | Excel 97-2003 Workbook |
| TextBoxInsertExcel | toggleButton | Text Box |
| FileSaveAsMenu | splitButton | Save As Other Format |
| FilePrintMenu | splitButton | Preview and Print |
| FilePermissionRestrictMenu | menu | Restrict Permission |
| InsertCellstMenu | splitButton | Insert |
| PivotTableInsertMenu | splitButton | PivotTable |
| OutlineGroupMenu | splitButton | Group |
| OutlineUngroupMenu | splitButton | Ungroup |
| FormulaMoreFunctionsMenu | menu | More Functions |
| DocumentPanelTemplate | button | Document Panel |
| GroupModify | group | Modify |
| ViewGridlinesToggleExcel | toggleButton | View Gridlines |
| BevelShapeGallery | gallery | Bevel |
| _3DBevelPictureTopGallery | gallery | Bevel |
| EditLinks | button | Edit Links |
| GlowColorPicker | gallery | More Glow Colors |
| RecolorColorPicker | gallery | More Variations |
| GlowColorMoreColorsDialog | button | More Colors... |

| idMso | Control Type | Label |
|-------------------------------------|--------------|---------------------------------------|
| PictureRecolorMoreColorsDialog | button | More Colors... |
| SmartArtAddBullet | button | Add Bullet |
| PictureChange | button | Change Picture... |
| GroupWordArtStyles | group | WordArt Styles |
| TextFillColorPicker | gallery | Text Fill |
| TextOutlineColorPicker | gallery | Text Outline |
| TextOutlineColorMoreColorsDialog | button | More Outline Colors... |
| TextEffectsMenu | menu | Text Effects |
| TextStylesGallery | gallery | Quick Styles |
| WordArtClear | button | Clear WordArt |
| TextPictureFill | button | Picture... |
| TextFillGradientGallery | gallery | Gradient |
| TextFillMoreGradientsDialog | button | More Gradients... |
| TextFillTextureGallery | gallery | Texture |
| TextOutlineDashesGallery | gallery | Dashes |
| TextOutlineMoreLinesDialog | button | More Lines... |
| TextOutlineWeightGallery | gallery | Weight |
| TextEffectShadowGallery | gallery | Shadow |
| TextEffectsMoreShadowsDialog | button | Shadow Options... |
| TextEffectsBevelMore3DOptionsDialog | button | 3-D Options... |
| TextEffects3DRotationGallery | gallery | 3-D Rotation |
| TextEffects3DRotationOptionsDialog | button | 3-D Rotation Options... |
| TextEffectGlowGallery | gallery | Glow |
| TextGlowColorPicker | gallery | More Glow Colors |
| TextGlowColorMoreColorsDialog | button | More Colors... |
| TextReflectionGallery | gallery | Reflection |
| ShapeEffectsMenu | menu | Shape Effects |
| PivotTableSubtotalsDoNotShow | button | Do Not Show Subtotals |
| PivotTableSubtotalsOnBottom | button | Show all Subtotals at Bottom of Group |

| idMso | Control Type | Label |
|---|---------------------|------------------------------------|
| PivotTableSubtotalsOnTop | button | Show all Subtotals at Top of Group |
| PivotTableGrandTotalsOffForRowsAndColumns | button | Off for Rows and Columns |
| PivotTableGrandTotalsOnForRowsAndColumns | button | On for Rows and Columns |
| PivotTableGrandTotalsOnForRowsOnly | button | On for Rows Only |
| PivotTableGrandTotalsOnForColumnsOnly | button | On for Columns Only |
| PivotTableBlankRowsInsert | button | Insert Blank Line after Each Item |
| PivotTableBlankRowsRemove | button | Remove Blank Line after Each Item |
| GroupChartCurrentSelection | group | Current Selection |
| GroupChartLabels | group | Labels |
| AlignTopExcel | toggleButton | Top Align |
| AlignMiddleExcel | toggleButton | Middle Align |
| AlignBottomExcel | toggleButton | Bottom Align |
| BevelTextGallery | gallery | Bevel |
| PictureCorrectionsDialog | button | Picture Corrections Options... |
| GroupTableStylesExcel | group | Table Styles |
| ConditionalFormattingNewRule | button | New Rule... |
| SmartArtAddShapeSplitMenu | splitButton | Add Shape Options |
| ViewRulerExcel | checkBox | Ruler |
| GroupInkFormat | group | Format |
| InkColorPicker | gallery | Color |
| BorderColorPickerExcel | gallery | Line Color |
| GroupSmartArtShapes | group | Shapes |
| GroupInsertText | group | Text |
| ShapeOutlineColorPicker | gallery | Picture Border |
| TableDeleteRowsAndColumnsMenu | splitButton | Delete |
| AccountingFormatMenu | splitButton | |
| GroupChartProperties | group | Properties |
| PivotTableEditDataSource | splitButton | Edit Data Source |
| FileExcelServicesOptions | button | Excel Services Options |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|---------------------------------------|
| TableExportMenu | menu | Export |
| TableExportTableToVisioPivotDiagram | button | Export Table to Visio PivotDiagram... |
| PasteMenu | splitButton | Paste |
| GroupPictureStyles | group | Picture Styles |
| PictureStylesGallery | gallery | Quick Styles |
| GroupInsertChartsExcel | group | Charts |
| ChartTypeColumnInsertGallery | gallery | Column |
| ChartTypeLineInsertGallery | gallery | Line |
| ChartTypePieInsertGallery | gallery | Pie |
| ChartTypeBarInsertGallery | gallery | Bar |
| ChartTypeAreaInsertGallery | gallery | Area |
| ChartTypeXYScatterInsertGallery | gallery | Scatter |
| ChartTypeOtherInsertGallery | gallery | Other Charts |
| ChartTypeAllInsertDialog | button | All Chart Types... |
| PivotChartClearMenu | menu | Clear |
| PictureEffectsShadowGallery | gallery | Shadow |
| PictureEffectsGlowGallery | gallery | Glow |
| PictureEffectsSoftEdgesGallery | gallery | Soft Edges |
| PictureReflectionGallery | gallery | Reflection |
| PictureRotationGallery | gallery | 3-D Rotation |
| InkToolsClose | button | Close Ink Tools |
| SheetRowsInsert | button | Insert Sheet Rows |
| SheetColumnsInsert | button | Insert Sheet Columns |
| SheetRowsDelete | button | Delete Sheet Rows |
| SheetColumnsDelete | button | Delete Sheet Columns |
| GroupViewShowHide | group | Show/Hide |
| GroupWindow | group | Window |
| GroupWorkbookViews | group | Workbook Views |
| ViewHeadings | checkBox | View |

| idMso | Control Type | Label |
|-------------------------------|---------------------|------------------------------|
| RefreshAllMenu | splitButton | Refresh |
| HideAndUnhideMenu | menu | Hide & Unhide |
| DataValidationMenu | splitButton | Data Validation |
| FileDocumentEncrypt | toggleButton | Encrypt Document |
| WordArtFormatDialog | button | Format Text Effects... |
| ObjectRotationOptionsDialog | button | More Rotation Options... |
| MoreTextureOptions | button | More Textures... |
| TextFillColorMoreColorsDialog | button | More Fill Colors... |
| BorderTopNoToggle | button | Top Border |
| BorderBottomNoToggle | button | Bottom Border |
| BorderLeftNoToggle | button | Left Border |
| BorderRightNoToggle | button | Right Border |
| WindowSplitToggle | toggleButton | Split |
| FileEmailAsPdfEmailAttachment | button | E-mail as PDF Attachment |
| FileEmailAsXpsEmailAttachment | button | E-mail as XPS Attachment |
| GroupPrintPreviewZoom | group | Zoom |
| PictureEffectsMenu | menu | Picture Effects |
| PictureShapeGallery | gallery | Change Shape |
| GroupChartBackground | group | Background |
| GroupChartAnalysis | group | Analysis |
| ZoomToSelection | button | Zoom to Selection |
| GridlinesExcel | checkBox | View |
| UnmergeCells | button | Split Cells... |
| MenuPublish | menu | Publish |
| ViewSideBySide | toggleButton | View Side by Side |
| FileSaveAsOtherFormats | button | Save As |
| FileSaveAsExcelXlsx | button | Excel Workbook |
| FileSaveAsExcelXlsxMacro | button | Excel Macro-Enabled Workbook |
| FileSaveAsExcelXlsb | button | Excel Binary Workbook |

| idMso | Control Type | Label |
|---------------------------|---------------------|-------------------------------------|
| PasteAsPictureMenu | menu | As Picture |
| GroupPivotActions | group | Actions |
| PivotTableSelectFlyout | menu | Select |
| ZoomCurrent100 | button | 100% |
| TextFillMoreTextures | button | More Textures... |
| GroupMacros | group | Macros |
| PlayMacro | button | Macros |
| MenuMacros | splitButton | Macros |
| AdvertisePublishAs | button | Find add-ins for other file formats |
| UpgradeWorkbook | button | Convert |
| ReviewProtectWorkbookMenu | menu | Protect Workbook |
| ReviewRestrictEditing | toggleButton | Protect Structure and Windows |
| AlternativeText | button | Size and Properties... |
| ThemeBrowseForThemes | button | Browse for Themes... |
| FileCheckOutDiscard | button | Discard Check Out |
| MdiChildSystemMenu | menu | Document |

3.1.3 PowerPoint 2007

| idMso | Control Type | Label |
|--------------|---------------------|-----------------|
| Spelling | button | Spelling... |
| FileSave | button | Save |
| FilePrint | button | Print |
| TableInsert | button | Insert Table... |
| ChartInsert | button | Chart... |
| FileNew | button | New |
| Copy | button | Copy |
| Cut | button | Cut |
| Paste | button | Paste |
| FileOpen | button | Open |

| idMso | Control Type | Label |
|------------------|---------------------|----------------|
| Clear | button | Clear |
| Superscript | toggleButton | Superscript |
| Subscript | toggleButton | Subscript |
| FileClose | button | Close |
| FormatPainter | toggleButton | Format Painter |
| FilePrintPreview | toggleButton | Print Preview |
| PickUpStyle | button | Pick Up Style |
| PasteApplyStyle | button | Apply Style |
| Bold | toggleButton | Bold |
| Italic | toggleButton | Italic |
| Underline | toggleButton | Underline |
| AlignLeft | toggleButton | Align Left |
| AlignRight | toggleButton | Align Right |
| AlignCenter | toggleButton | Center |
| AlignJustify | toggleButton | Justify |
| Undo | gallery | Undo |
| Redo | gallery | Redo |
| OutlinePromote | button | Promote |
| OutlineDemote | button | Demote |
| OutlineMoveUp | button | Move Up |
| OutlineMoveDown | button | Move Down |
| OutlineExpand | button | Expand |
| OutlineCollapse | button | Collapse |
| TextBoxInsert | toggleButton | Text Box |
| FindDialog | button | Find... |
| BorderTop | toggleButton | Top Border |
| BorderBottom | toggleButton | Bottom Border |
| BorderLeft | toggleButton | Left Border |
| BorderRight | toggleButton | Right Border |

| idMso | Control Type | Label |
|---------------------------|---------------------|----------------------|
| BorderInside | toggleButton | Inside Borders |
| BorderOutside | toggleButton | Outside Borders |
| BorderNone | toggleButton | No Border |
| ObjectsGroup | button | Group |
| ObjectsUngroup | button | Ungroup |
| ObjectBringToFront | button | Bring to Front |
| ObjectSendToBack | button | Send to Back |
| ObjectBringForward | button | Bring Forward |
| ObjectSendBackward | button | Send Backward |
| ViewFullScreenView | button | Full Screen |
| ViewRulerPowerPoint | checkBox | Ruler |
| ObjectsSelect | toggleButton | Select Objects |
| MacroPlay | button | Macros |
| ObjectFlipHorizontal | button | Flip Horizontal |
| ObjectFlipVertical | button | Flip Vertical |
| ObjectRotateRight90 | button | Rotate Right 90° |
| ObjectRotateLeft90 | button | Rotate Left 90° |
| GroupDrawing | group | Drawing |
| ObjectEditPoints | toggleButton | Edit Points |
| GridSettings | button | Grid Settings... |
| PropertySheet | button | Property Sheet |
| OutlineShowTextFormatting | toggleButton | Show Text Formatting |
| Strikethrough | toggleButton | Strikethrough |
| WindowsArrangeAll | button | Arrange All |
| WindowNew | button | New Window |
| SymbolInsert | button | Symbol... |
| ReplaceDialog | button | Replace... |
| PagePrevious | button | Previous Page |
| PageNext | button | Next Page |

| idMso | Control Type | Label |
|--------------------------|---------------------|--|
| TextBoxInsertVertical | toggleButton | Vertical Text Box |
| RedoOrRepeat | button | Redo |
| ObjectsRegroup | button | Regroup |
| FontSizeIncrease | button | Increase Font Size |
| FontSizeDecrease | button | Decrease Font Size |
| OleObjectctInsert | button | Object... |
| SnapToGrid | toggleButton | Snap to Grid |
| FindNext | button | Find Next |
| PasteDuplicate | button | Duplicate |
| SlideNew | button | New Slide |
| ClipArtInsert | toggleButton | Clip Art... |
| CreateHandoutsInWord | button | Create Handouts in Microsoft Office Word |
| Shadow | toggleButton | Shadow |
| ObjectRotateFree | button | Free Rotate |
| ShapesMoreShapes | button | More AutoShapes |
| SlideMasterMasterLayout | button | Slide Layout... |
| CollapseAll | button | Collapse All |
| OutlineExpandAll | button | Expand All |
| CombineCharacters | toggleButton | Yoko-Gumi |
| SlideHide | toggleButton | Hide Slide |
| AnimationCustom | toggleButton | Custom Animation... |
| PictureCrop | toggleButton | Crop |
| SlideShowRehearseTimings | button | Rehearse Timings |
| ViewSlideView | toggleButton | Slide |
| ViewOutlineView | toggleButton | Outline |
| ViewSlideSorterView | toggleButton | Slide Sorter |
| ViewNotesPageView | toggleButton | Notes Page |
| ViewSlideShowView | button | Slide Show |
| ViewSlideMasterView | toggleButton | Slide Master |

| idMso | Control Type | Label |
|----------------------------------|---------------------|--------------------------------------|
| FileSaveAs | button | Save As |
| AdvancedFileProperties | button | View Document Properties... |
| PasteSpecialDialog | button | Paste Special... |
| SelectAll | button | Select All |
| FileLinksToFiles | button | Edit Links to Files |
| HeaderFooterInsert | button | Header & Footer... |
| DateAndTimeInsert | button | Date & Time... |
| NumberInsert | button | Number... |
| BordersShadingDialog | button | Borders and Shading... |
| BulletsAndNumberingBulletsDialog | button | Bullets and Numbering... |
| SetLanguage | button | Set Language... |
| AutoCorrect | button | AutoCorrect Options... |
| MergeCells | button | Merge Cells |
| SplitCells | button | Split Cells... |
| TableRowSelect | button | Select Row |
| TableColumnSelect | button | Select Column |
| TableSelect | button | Select Table |
| ShowClipboard | button | Office Clipboard... |
| OutlookTaskCreate | button | Create Microsoft Office Outlook Task |
| WindowMinimize | button | Minimize |
| WindowRestore | button | Restore |
| WindowClose | button | Close |
| PrintPreviewClose | button | Close Print Preview |
| ZoomDialog | button | Zoom... |
| About | button | About |
| PictureInsertFromFilePowerPoint | button | Picture... |
| ExchangeFolder | button | Exchange Folder... |
| AddInManager | button | Add-Ins... |
| ChartEditDataSource | button | Select Data... |

| idMso | Control Type | Label |
|--|---------------------|------------------------|
| WindowMoreWindowsDialog | toggleButton | More Windows... |
| ObjectEditDialog | button | Object... |
| ObjectFormatDialog | button | Object... |
| Help | button | Help |
| WebGoBack | button | Back |
| WebGoForward | button | Forward |
| SmartArtInsert | button | SmartArt... |
| ShapeRerouteConnectors | toggleButton | Reroute Connectors |
| ObjectNudgeUp | button | Up |
| ObjectNudgeDown | button | Down |
| ObjectNudgeLeft | button | Left |
| ObjectNudgeRight | button | Right |
| ShapeStraightConnector | toggleButton | Straight Connector |
| ShapeElbowConnector | toggleButton | Elbow Connector |
| ObjectFillMoreColorsDialog | button | More Fill Colors... |
| ObjectBorderOutlineColorMoreColorsDialog | button | More Outline Colors... |
| LineStyleDialog | button | More Lines... |
| ArrowsMore | button | More Arrows... |
| WordArtVerticalText | button | Vertical Text |
| ContrastMore | button | More Contrast |
| ContrastLess | button | Less Contrast |
| BrightnessMore | button | More Brightness |
| BrightnessLess | button | Less Brightness |
| ShadowNudgeUpClassic | button | Nudge Shadow Up |
| ShadowNudgeDownClassic | button | Nudge Shadow Down |
| ShadowNudgeLeftClassic | button | Nudge Shadow Left |
| ShadowNudgeRightClassic | button | Nudge Shadow Right |
| ShapeRectangle | toggleButton | Rectangle |
| ShapeRoundedRectangle | toggleButton | Rounded Rectangle |

| idMso | Control Type | Label |
|--------------------------------|---------------------|-----------------------------|
| ShapeIsoscelesTriangle | toggleButton | Isosceles Triangle |
| ShapeOval | toggleButton | Oval |
| ShapeLeftBrace | toggleButton | Left Brace |
| ShapeRightBrace | toggleButton | Right Brace |
| ShapeArc | toggleButton | Arc |
| ShapeRightArrow | toggleButton | Right Arrow |
| ShapeDownArrow | toggleButton | Down Arrow |
| ShapeRoundedRectangularCallout | toggleButton | Rounded Rectangular Callout |
| ShapeStar | toggleButton | 5-Point Star |
| PictureReset | button | Reset Picture |
| SnapToShapes | toggleButton | Snap to Shape |
| ViewVisualBasicCode | button | View Code |
| MasterViewClose | button | Close |
| HyperlinkInsert | button | Hyperlink... |
| ReviewNewComment | button | New Comment |
| VisualBasic | button | Visual Basic |
| BordersAll | toggleButton | All Borders |
| SlideBackgroundFormatDialog | button | Format Background... |
| TableDrawBorderPenStyle | dropDown | Pen Style |
| Font | comboBox | Font: |
| FontSize | comboBox | Font Size: |
| ZoomClassic | gallery | Zoom: |
| DocumentLocation | comboBox | Address: |
| InsertTab | button | Tab |
| WindowsCascade | button | Cascade |
| BorderInsideHorizontal | toggleButton | Inside Horizontal Border |
| BorderInsideVertical | toggleButton | Inside Vertical Border |
| BorderDiagonalDown | toggleButton | Diagonal Down Border |
| BorderDiagonalUp | toggleButton | Diagonal Up Border |

| idMso | Control Type | Label |
|--------------------------------|---------------------|------------------------|
| TextDirectionLeftToRight | toggleButton | Left-to-Right |
| TextDirectionRightToLeft | toggleButton | Right-to-Left |
| ActiveXCheckBox | button | Check Box |
| ActiveXTextBox | button | Text Box |
| ActiveXButton | button | Command Button |
| ActiveXRadioButton | button | Option Button |
| ActiveXListBox | button | List Box |
| ActiveXComboBox | button | Combo Box |
| ActiveXToggleButton | button | Toggle Button |
| ActiveXSpinButton | button | Spin Button |
| ActiveXScrollBar | button | Scroll Bar |
| ActiveXLabel | button | Label |
| OleConvert | button | Convert... |
| ReviewEditComment | button | Edit Comment |
| TableDrawTable | toggleButton | Draw Table |
| TableEraser | toggleButton | Eraser |
| TableCellAlignTop | toggleButton | Align Top |
| TableCellAlignCenterVertically | toggleButton | Center Vertically |
| TableCellAlignBottom | toggleButton | Align Bottom |
| TableColumnsDistribute | button | Distribute Columns |
| TableRowsDistribute | button | Distribute Rows |
| ActiveXImage | button | Image |
| TableDeleteRows | button | Delete Rows |
| TableDeleteColumns | button | Delete Columns |
| ShadowOnOrOffClassic | button | Shadow On/Off |
| ObjectSetShapeDefaults | button | Set AutoShape Defaults |
| FileSendAsAttachment | button | E-mail |
| FileNewDefault | button | New |
| FilePrintQuick | button | Quick Print |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|---------------------------------|
| PictureInsertFromFile | button | Picture... |
| TableDrawBorderPenWeight | dropDown | Pen Weight |
| TableShowGridlines | toggleButton | View Gridlines |
| TableBorderPenColorPicker | gallery | Pen Color |
| ShapeStraightConnectorArrow | toggleButton | Straight Arrow Connector |
| ShapeElbowConnectorArrow | toggleButton | Elbow Arrow Connector |
| SlideDelete | button | Delete |
| ViewHandoutMasterView | toggleButton | Handout Master |
| ViewNotesMasterView | toggleButton | Notes Master |
| SlidesReuseSlides | toggleButton | Reuse Slides... |
| SlidesFromOutline | button | Slides from Outline... |
| MovieFromClipOrganizerInsert | button | Movie from Organizer... |
| MovieFromFileInsert | button | Movie from File... |
| SoundInsertFromClipOrganizer | button | Sound from Clip Organizer... |
| SoundInsertFromFile | button | Sound from File... |
| CDAudioPlayTrack | button | Play CD Audio Track... |
| FontsReplaceFonts | button | Replace Fonts... |
| ChangeCaseToggle | button | Toggle Case |
| BlackAndWhiteAutomatic | toggleButton | B & W Automatic |
| BlackAndWhiteBlack | toggleButton | B & W |
| BlackAndWhiteBlackWithGrayscaleFill | toggleButton | B & W Black with Grayscale Fill |
| BlackAndWhiteBlackWithWhiteFill | toggleButton | B & W Black with White Fill |
| BlackAndWhiteDontShow | toggleButton | B & W Don't Show |
| BlackAndWhiteGrayWithWhiteFill | toggleButton | B & W Gray with White Fill |
| BlackAndWhiteGrayscale | toggleButton | B & W Grayscale |
| BlackAndWhiteInverseGrayscale | toggleButton | B & W Inverse Grayscale |
| BlackAndWhiteWhite | toggleButton | B & W White |
| BlackAndWhiteLightGrayscale | toggleButton | B & W Light Grayscale |
| RecordNarration | button | Record Narration |

| idMso | Control Type | Label |
|---------------------------------|---------------------|----------------------------|
| SlideShowSetUpDialog | button | Set Up Slide Show... |
| PasteAsHyperlink | button | Paste as Hyperlink |
| ParagraphDistributed | toggleButton | Distributed |
| CharacterFormattingReset | button | Reset Character Formatting |
| PictureSetTransparentColor | toggleButton | Set Transparent Color |
| SlideShowCustom | button | Custom Slide Show |
| WindowFitToPage | button | Fit to Page |
| SoundRecord | button | Record Sound... |
| DuplicateSelectedSlides | button | Duplicate Selected Slides |
| ActionInsert | button | Action |
| SlideShowPreviousSlide | button | Previous |
| SlideShowNextSlide | button | Next |
| LinkBreak | button | Break Link |
| LinkChange | button | Change Link |
| LinksUpdate | button | Update Links |
| BaselineLower | button | Lower Baseline |
| BaselineRaise | button | Raise Baseline |
| SlideShowInAWindow | button | Slide Show in a Window |
| HangulHanjaConversionPowerPoint | button | Hangul Hanja Conversion |
| SlidesPerPage2Slides | toggleButton | 2 Slides |
| SlidesPerPage3Slides | toggleButton | 3 Slides |
| SlidesPerPage6Slides | toggleButton | 6 Slides |
| SlidesPerPageSlideOutline | toggleButton | Slide Outline |
| GoToProperty | button | Go To... |
| IndentIncrease | button | Increase Indent |
| IndentDecrease | button | Decrease Indent |
| MoviePlay | button | Preview |
| SlidesPerPage4Slides | toggleButton | 4 Slides |
| SlidesPerPage9Slides | toggleButton | 9 Slides |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|-------------------------|
| MacroSecurity | button | Macro Security |
| WebPagePreview | button | Web Page Preview |
| TableInsertDialog | button | Table... |
| TableInsertRowsAbove | button | Insert Above |
| TableInsertRowsBelow | button | Insert Below |
| TableInsertColumnsLeft | button | Insert Left |
| TableInsertColumnsRight | button | Insert Right |
| ComAddInsDialog | button | COM Add-Ins... |
| ViewNormalViewPowerPoint | toggleButton | Normal |
| WebOptionsDialog | button | Web Options... |
| ViewDirectionLeftToRight | toggleButton | Left-to-Right |
| ViewDirectionRightToLeft | toggleButton | Right-to-Left |
| NextPane | button | Next Pane |
| WindowMoveSplit | button | Move Split |
| TableSetLeftToRight | toggleButton | Set Left-to-Right Table |
| TableSetRightToLeft | toggleButton | Set Right-to-Left Table |
| SlideShowResumeShow | button | Resume Slide Show |
| ObjectsMultiSelect | button | Select Multiple Objects |
| FontColorMoreColorsDialogPowerPoint | button | More Colors... |
| DrawingCanvasFit | button | Fit |
| DrawingCanvasResize | button | Resize |
| ViewBackToColorView | toggleButton | Back To Color View |
| ViewDisplayInGrayscale | toggleButton | Grayscale |
| ViewDisplayInHighContrast | toggleButton | High Contrast |
| ViewDisplayInPureBlackAndWhite | toggleButton | Pure Black and White |
| PrintWhat | dropDown | Print What |
| DrawingCanvasExpand | button | Expand |
| SlidesPerPage1Slide | toggleButton | 1 Slide |
| SlideMasterPreserveMaster | toggleButton | Preserve |

| idMso | Control Type | Label |
|------------------------------------|---------------------|--|
| SlideMasterRenameMaster | button | Rename |
| OutlineThumbnailsShowHide | button | Show Outline |
| BulletsAndNumberingNumberingDialog | button | Bullets and Numbering... |
| FileCheckOut | button | Check Out |
| FileCheckIn | button | Check In |
| PageOrientationLandscape | toggleButton | Landscape |
| PageOrientationPortrait | toggleButton | Portrait |
| SlideReset | button | Reset |
| PicturesCompress | button | Compress Pictures... |
| ReviewDeleteCommentPowerPoint | button | Delete |
| ReviewNextCommentPowerPoint | button | Next |
| ReviewPreviousCommentPowerPoint | button | Previous |
| ReviewShowOrHideMarkup | toggleButton | Show Markup |
| ReviewDeleteAllMarkupOnSlide | button | Delete All Markup on the Current Slide |
| Translate | button | Translate... |
| ViewGridlinesPowerPoint | checkBox | View Gridlines |
| PhotoAlbumInsert | button | New Photo Album... |
| PhotoAlbumEdit | button | Edit Photo Album... |
| DrawingCanvasScale | button | Scale Drawing |
| ResearchPane | toggleButton | Research... |
| FileInternetFax | button | Internet Fax |
| FilePackageForCD | button | Package for CD |
| FileVersionHistory | button | View Version History |
| InkColorMoreColorsDialog | button | More Ink Colors... |
| ContactUs | button | Contact Us... |
| FilePermissionUnrestricted | toggleButton | Unrestricted Access |
| FilePermissionDoNotDistribute | toggleButton | Restricted Access |
| FilePermissionView | button | View Permission |
| Thesaurus | button | Thesaurus... |

| idMso | Control Type | Label |
|--------------------------------|---------------------|----------------------|
| InkingStart | button | Start Inking |
| CheckForUpdates | button | Check for Updates |
| InkEraser | toggleButton | Eraser |
| FilePermissionRestrictAs | button | Manage Credentials |
| ThemeBrowseForThemesPowerPoint | button | Browse for Themes... |
| SlideMasterInsertLayout | button | Insert Layout |
| FileViewDigitalSignatures | toggleButton | View Signatures |
| FileWorkflowTasks | button | View Workflow Tasks |
| FileStartWorkflow | button | Workflows |
| SlideThemesGallery | gallery | Themes |
| LabelInsert | button | Label |
| BarcodeInsert | button | Barcode |
| ChartStylesGallery | gallery | Quick Styles |
| ChartLayoutGallery | gallery | Quick Layout |
| ChartSaveTemplates | button | Save As Template |
| ChartAxisTitles | menu | Axis Titles |
| ChartAxes | menu | Axes |
| ChartGridlines | menu | Gridlines |
| ChartFormatSelection | button | Format Selection |
| ChartElementSelector | comboBox | Chart Elements |
| SlideShowFromCurrent | button | From Current Slide |
| BulletsGallery | gallery | Bullets |
| NumberingGallery | gallery | Numbering |
| LineSpacingGalleryPowerPoint | gallery | Line Spacing |
| SlideNewGallery | gallery | New Slide |
| SlideLayoutGallery | gallery | Layout |
| TabPictureToolsFormat | tab | Format |
| TabDrawingToolsFormat | tab | Format |
| ShapesInsertGallery | gallery | Shapes |

| idMso | Control Type | Label |
|----------------------------|---------------------|--------------------------|
| ShapeChangeShapeGallery | gallery | Change Shape |
| ShapeFillTextureGallery | gallery | Texture |
| ShapeStylesGallery | gallery | Quick Styles |
| PageOrientationGallery | menu | Orientation |
| FileServerTasksMenu | menu | Server |
| FileSendMenu | menu | Send |
| TabInsert | tab | Insert |
| TabView | tab | View |
| GroupFont | group | Font |
| GroupParagraph | group | Paragraph |
| GroupProofing | group | Proofing |
| GroupInsertIllustrations | group | Illustrations |
| GroupShapes | group | Insert Shapes |
| GroupPageSetup | group | Page Setup |
| GroupComments | group | Comments |
| GroupPictureSize | group | Size |
| GroupDrawBorders | group | Draw Borders |
| TabDesign | tab | Design |
| TabAnimations | tab | Animations |
| TabSlideShow | tab | Slide Show |
| GroupSlides | group | Slides |
| GroupArrange | group | Arrange |
| GroupInsertMediaClips | group | Media Clips |
| GroupSlideThemes | group | Themes |
| GroupBackground | group | Background |
| GroupPreview | group | Preview |
| GroupAnimations | group | Animations |
| GroupTransitionToThisSlide | group | Transition to This Slide |
| GroupSlideShowStart | group | Start Slide Show |

| idMso | Control Type | Label |
|-----------------------------|---------------------|--------------------|
| GroupSlideShowSetup | group | Set Up |
| GroupShapeStyles | group | Shape Styles |
| TabReview | tab | Review |
| GroupChartLayouts | group | Chart Layouts |
| GroupChartStyles | group | Chart Styles |
| GroupChartAxes | group | Axes |
| GroupChartShapes | group | Insert |
| ObjectEffectPresetGallery | gallery | Preset |
| PictureEffectsPresetGallery | gallery | Preset |
| _3DRotationGallery | gallery | 3-D Rotation |
| TabSmartArtToolsDesign | tab | Design |
| TabSmartArtToolsFormat | tab | Format |
| TabChartToolsDesign | tab | Design |
| TabChartToolsLayout | tab | Layout |
| TabChartToolsFormat | tab | Format |
| ShapeFillColorPicker | gallery | Shape Fill |
| OutlineColorPicker | gallery | Picture Border |
| FileDocumentInspect | button | Inspect Document |
| ClearFormatting | button | Clear Formatting |
| GroupControls | group | Controls |
| GroupZoom | group | Zoom |
| FilePublishSlides | button | Publish Slides |
| SlideShowUsePresenterView | checkBox | Use Presenter View |
| ArrowStyleGallery | gallery | Arrows |
| OutlineDashesGallery | gallery | Dashes |
| OutlineWeightGallery | gallery | Weight |
| TabTableToolsLayout | tab | Layout |
| GroupAlignment | group | Alignment |
| GroupPictureTools | group | Adjust |

| idMso | Control Type | Label |
|---------------------------------|---------------------|-------------------------------|
| GroupSize | group | Size |
| TabSlideMaster | tab | Slide Master |
| GroupTableStylesPowerPoint | group | Table Styles |
| GroupTableRowsAndColumns | group | Rows & Columns |
| ObjectAlignMenu | menu | Align |
| MovieInsert | splitButton | Movie |
| SoundInsertMenu | splitButton | Sound |
| ObjectRotateGallery | gallery | Rotate |
| SelectMenu | menu | Select |
| AnimationGallery | dropDown | Animate: |
| AnimationTransitionGallery | gallery | Transition Scheme |
| AnimationTransitionSpeedGallery | dropDown | Transition Speed: |
| FontColorPicker | gallery | Font Color |
| TableColumnsGallery | gallery | Columns |
| TabHome | tab | Home |
| ChartTitle | gallery | Chart Title |
| ChartPrimaryHorizontalAxisTitle | gallery | Primary Horizontal Axis Title |
| ChartPrimaryVerticalAxisTitle | gallery | Primary Vertical Axis Title |
| ChartDepthAxisTitle | gallery | Depth Axis Title |
| ChartLegend | gallery | Legend |
| ChartDataLabel | gallery | Data Labels |
| ChartPrimaryHorizontalGridlines | gallery | Primary Horizontal Gridlines |
| ChartPrimaryVerticalGridlines | gallery | Primary Vertical Gridlines |
| ChartDepthGridlines | gallery | Depth Gridlines |
| ChartPrimaryHorizontalAxis | gallery | Primary Horizontal Axis |
| ChartPrimaryVerticalAxis | gallery | Primary Vertical Axis |
| ChartDepthAxis | gallery | Depth Axis |
| ChartDataTable | gallery | Data Table |
| ChartTrendline | gallery | Trendline |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|---|
| ChartErrorBars | gallery | Error Bars |
| ChartLines | gallery | Lines |
| ChartUpDownBars | gallery | Up/Down Bars |
| ChartPlotArea | gallery | Plot Area |
| ChartWall | gallery | Chart Wall |
| ChartFloor | gallery | Chart Floor |
| SmartArtAddShape | button | Add Shape |
| SmartArtLargerShape | button | Larger |
| SmartArtSmallerShape | button | Smaller |
| SmartArtResetGraphic | button | Reset Graphic |
| SmartArtTextPane | toggleButton | Text Pane |
| SmartArtEditIn2D | toggleButton | Edit in 2-D |
| SmartArtLayoutGallery | gallery | Change Layout |
| SmartArtMoreLayoutsDialog | button | More Layouts... |
| SmartArtStylesGallery | gallery | Quick Styles |
| SmartArtChangeColorsGallery | gallery | Change Colors |
| ObjectEffectSoftEdgesGallery | gallery | Soft Edges |
| ObjectEffectGlowGallery | gallery | Glow |
| GradientGallery | gallery | Gradient |
| ObjectEffectShadowGallery | gallery | Shadow |
| WordArtInsertGallery | gallery | WordArt |
| TextEffectTransformGallery | gallery | Transform |
| ThemeColorsGallery | gallery | Colors |
| GroupSmartArtLayouts | group | Layouts |
| GroupSmartArtQuickStyles | group | SmartArt Styles |
| GroupSmartArtCreateGraphic | group | Create Graphic |
| GroupSmartArtReset | group | Reset |
| GroupSmartArtSize | group | Size |
| ThemeSearchOfficeOnlinePowerPoint | button | More Themes on Microsoft Office Online... |

| idMso | Control Type | Label |
|--|---------------------|---------------------------|
| TabAddIns | tab | Add-Ins |
| ObjectEditShapeMenu | menu | Edit Shape |
| SlideMasterContentPlaceholderInsert | button | Content |
| SlideMasterTextPlaceholderInsert | button | Text |
| SlideMasterChartPlaceholderInsert | button | Chart |
| SlideMasterTablePlaceholderInsert | button | Table |
| SlideMasterDiagramPlaceholderInsert | button | SmartArt |
| SlideMasterMediaPlaceholderInsert | button | Media |
| SlideMasterClipArtPlaceholderInsert | button | Clip Art |
| SlideMasterVerticalTextPlaceholderInsert | button | Text |
| SlideMasterShowTitle | checkBox | Title |
| SlideMasterShowFooters | checkBox | Footers |
| TableStylesGallery | gallery | Table Styles |
| FileSaveAsPdfOrXps | button | Publish as PDF or XPS |
| FileSaveAsPowerPointOpenDocumentPresentation | button | OpenDocument Presentation |
| MoreControlsDialog | button | More Controls... |
| GroupCode | group | Code |
| TabDeveloper | tab | Developer |
| SlideTransitionApplyToAll | button | Apply To All |
| SlideTransitionOnMouseClick | checkBox | On Mouse Click |
| AnimationPreview | button | Preview |
| GroupEditing | group | Editing |
| GroupClipboard | group | Clipboard |
| GroupInsertTables | group | Tables |
| GroupInsertLinks | group | Links |
| GroupInsertBarcode | group | Barcode |
| AnimationTransitionSoundGallery | dropDown | Transition Sound: |
| ObjectPictureFill | button | Picture... |
| SlideShowFromBeginning | button | From Beginning |

| idMso | Control Type | Label |
|--------------------------------------|---------------------|----------------------------|
| WindowSwitchWindowsMenuPowerPoint | menu | Switch Windows |
| ThemeColorsCreateNew | button | Create New Theme Colors... |
| ThemeFontsCreateNew | button | Create New Theme Fonts... |
| ShapeFillMoreGradientsDialog | button | More Gradients... |
| ShadowOptionsDialog | button | Shadow Options... |
| ObjectsAlignSelectedSmart | toggleButton | Align Selected Objects |
| ObjectsAlignRelativeToContainerSmart | toggleButton | Align to Slide |
| ObjectsAlignLeftSmart | button | Align Left |
| ObjectsAlignRightSmart | button | Align Right |
| ObjectsAlignTopSmart | button | Align Top |
| ObjectsAlignBottomSmart | button | Align Bottom |
| ObjectsAlignCenterHorizontalSmart | button | Align Center |
| ObjectsAlignMiddleVerticalSmart | button | Align Middle |
| AlignDistributeHorizontally | button | Distribute Horizontally |
| AlignDistributeVertically | button | Distribute Vertically |
| SlideShowCustomMenu | menu | Custom Slide Show |
| SlideShowResolutionGallery | dropDown | Resolution: |
| SlideShowUseRehearsedTimings | checkBox | Use Rehearsed Timings |
| SlideShowShowPresentationOnGallery | dropDown | Show Presentation On: |
| GroupMonitors | group | Monitors |
| ThemeFontsGallery | gallery | Fonts |
| ThemeEffectsGallery | gallery | Effects |
| FileProperties | toggleButton | Properties |
| TabPrintPreview | tab | Print Preview |
| GroupPrintPreviewPrint | group | Print |
| GroupPrintPreviewPreview | group | Preview |
| TabSoundToolsOptions | tab | Options |
| TabMovieToolsOptions | tab | Options |
| GroupPlay | group | Play |

| idMso | Control Type | Label |
|------------------------------------|---------------------|-----------------------------------|
| GroupMovieOptions | group | Movie Options |
| GroupSoundOptions | group | Sound Options |
| MediaClipToolsHideDuringShow | checkBox | Hide During Show |
| MediaClipLoopUntilStopped | checkBox | Loop Until Stopped |
| SoundPlaySoundGallery | dropDown | Play Sound: |
| SlideShowVolume | gallery | Slide Show Volume |
| MoviePlayFullScreen | checkBox | Play Full Screen |
| MovieRewindAfterPlaying | checkBox | Rewind Movie After Playing |
| TableBordersMenu | splitButton | Borders |
| FileCreateDocumentWorkspace | toggleButton | Create Document Workspace |
| FileSaveToDocumentManagementServer | button | Document Management Server |
| FileDocumentManagementInformation | toggleButton | Document Management Information |
| QuickAccessToolBarCustomization | button | Customize Quick Access Toolbar... |
| FilePrepareMenu | menu | Prepare |
| FileMarkAsFinal | toggleButton | Mark as Final |
| FileAddDigitalSignature | button | Add a Digital Signature |
| SlideBackgroundHideGraphics | checkBox | Hide Background Graphics |
| ChangeCaseGallery | gallery | Change Case |
| ReviewDeleteCommentsMenuPowerPoint | splitButton | Delete |
| GroupMerge | group | Merge |
| AlignJustifyMenu | menu | Justify |
| TextBoxInsertMenu | splitButton | Text Box |
| TextBoxInsertHorizontal | toggleButton | Horizontal Text Box |
| ControlProperties | button | Properties |
| HandoutOrientation | gallery | Handout Orientation |
| SlidesPerPageGallery | menu | Slides Per Page |
| MasterShowSlideImage | checkBox | Slide Image |
| MasterShowBody | checkBox | Body |
| MasterShowDate | checkBox | Date |

| idMso | Control Type | Label |
|--------------------------------|---------------------|--------------------------|
| MasterShowPageNumber | checkBox | Page Number |
| MasterShowHeader | checkBox | Header |
| MasterShowFooter | checkBox | Footer |
| TabHandoutMaster | tab | Handout Master |
| TabNotesMaster | tab | Notes Master |
| GroupPlaceholdersHandoutMaster | group | Placeholders |
| GroupMasterEditTheme | group | Edit Theme |
| GroupMasterClose | group | Close |
| GroupMasterEdit | group | Edit Master |
| GroupMasterLayout | group | Master Layout |
| TabGrayscale | tab | Grayscale |
| TabBlackAndWhite | tab | Black And White |
| GroupColorModeSetting | group | Change Selected Object |
| GroupInkSelect | group | Select |
| PictureBrightnessGallery | gallery | Brightness |
| PictureContrastGallery | gallery | Contrast |
| ShadingColorPicker | gallery | Shading |
| SmartArtAddShapeAfter | button | Add Shape After |
| SmartArtAddShapeBefore | button | Add Shape Before |
| SmartArtAddShapeAbove | button | Add Shape Above |
| SmartArtAddShapeBelow | button | Add Shape Below |
| SmartArtAddAssistant | button | Add Assistant |
| ThemeColorsReset | button | Reset Slide Theme Colors |
| ChartSwitchRowColumn | button | Switch Row/Column |
| ChartShowData | button | Edit Data... |
| ChartRefresh | button | Refresh Data |
| ChartChangeType | button | Change Chart Type... |
| GroupChartData | group | Data |
| GroupChartType | group | Type |

| idMso | Control Type | Label |
|---|---------------------|-------------------------|
| GroupTableSize | group | Table Size |
| TableCellMarginsGallery | gallery | Cell Margins |
| SlideMasterVerticalContentPlaceholderInsert | button | Content (Vertical) |
| _3DRotationOptionsDialog | button | 3-D Rotation Options... |
| _3DBevelOptionsDialog | button | 3-D Options... |
| SlideBackgroundStylesGallery | gallery | Background Styles |
| TextDirectionGallery | gallery | Text Direction |
| GroupTableStyleOptionsPowerPoint | group | Table Style Options |
| TableStyleFirstRowPowerPoint | checkBox | Header Row |
| TableStyleFirstColumnPowerPoint | checkBox | First Column |
| TableStyleTotalRowPowerPoint | checkBox | Total Row |
| TableStyleLastColumnPowerPoint | checkBox | Last Column |
| TableStyleBandedRowsPowerPoint | checkBox | Banded Rows |
| TableStyleBandedColumnsPowerPoint | checkBox | Banded Columns |
| SelectionPane | toggleButton | Selection Pane... |
| SmartArtOrganizationChartLeftHanging | button | Left Hanging |
| SmartArtOrganizationChartRightHanging | button | Right Hanging |
| SmartArtOrganizationChartBoth | button | Both |
| SmartArtOrganizationChartStandard | button | Standard |
| SmartArtRightToLeft | toggleButton | Right to Left |
| TableCellCustomMarginsDialog | button | Custom Margins... |
| ViewMessageBar | checkBox | Message Bar |
| ShapeStylesOtherThemeFillsGallery | gallery | Other Theme Fills |
| SmartArtOrganizationChartMenu | menu | Layout |
| TabSlideMasterHome | tab | Home |
| TableInsertGallery | gallery | Table |
| AlignJustifyWithMixedLanguages | toggleButton | Justify |
| AlignJustifyLow | toggleButton | Justify Low |
| AlignJustifyThai | toggleButton | Distribute |

| idMso | Control Type | Label |
|---|---------------------|---------------------------|
| ExcelSpreadsheetInsert | button | Excel Spreadsheet |
| TabCDAudioToolsOptions | tab | Options |
| SlideTransitionAutomaticallyAfter | checkBox | Automatically After: |
| SlideOrientationGallery | gallery | Slide Orientation |
| GroupAddInsMenuCommands | group | Menu Commands |
| GroupAddInsToolbarCommands | group | Toolbar Commands |
| GroupInk | group | Ink |
| TabInkToolsPens | tab | Pens |
| GroupInkPens | group | Pens |
| GroupInkClose | group | Close |
| InkBallpointPen | toggleButton | Ballpoint Pen |
| InkFeltTipPen | toggleButton | Felt Tip Pen |
| InkHighlighter | toggleButton | Highlighter |
| ThemeSaveCurrentPowerPoint | button | Save Current Theme... |
| TextAlignGallery | gallery | Align Text |
| CharacterSpacingGallery | gallery | Character Spacing |
| ChartResetToMatchStyle | button | Reset to Match Style |
| Chart3DView | button | 3-D Rotation... |
| ObjectSizeAndPositionDialog | button | Size and Position... |
| ShapeConvertToFreeform | button | Convert to Freeform |
| TabTableToolsDesign | tab | Design |
| ConvertToSmartArt | gallery | Convert to SmartArt |
| ConvertToSmartArtMoreSmartArtGraphicsDialog | button | More SmartArt Graphics... |
| ReflectionGallery | gallery | Reflection |
| PictureRecolorGallery | gallery | Recolor |
| SmartArtPromote | button | Promote |
| SmartArtDemote | button | Demote |
| TableEffectsCellBevelGallery | gallery | Cell Bevel |
| TableBackgroundGallery | gallery | Table Background |

| idMso | Control Type | Label |
|--|--------------|---|
| ChartTitleOptionsDialog | button | More Title Options... |
| ChartLegendOptionsDialogDialog | button | More Legend Options... |
| ChartDataLabelDialog | button | More Data Label Options... |
| ChartPrimaryHorizontalAxisTitleOptionsDialog | button | More Primary Horizontal Axis Title Options... |
| ChartPrimaryVerticalAxisTitleOptionsDialog | button | More Primary Vertical Axis Title Options... |
| ChartSecondaryHorizontalAxisTitleOptionsDialog | button | More Secondary Horizontal Axis Title Options... |
| ChartSecondaryVerticalAxisTitleOptionsDialog | button | More Secondary Vertical Axis Title Options... |
| ChartDepthAxisTitleOptionsDialog | button | More Depth Axis Title Options... |
| ChartPrimaryHorizontalGridlinesOptionsDialog | button | More Primary Horizontal Gridlines Options... |
| ChartPrimaryVerticalGridlinesOptionsDialog | button | More Primary Vertical Gridlines Options... |
| ChartSecondaryHorizontalGridlinesOptionsDialog | button | More Secondary Horizontal Gridlines Options... |
| ChartSecondaryVerticalGridlinesOptionsDialog | button | More Secondary Vertical Gridlines Options... |
| ChartDepthGridlinesOptionsDialog | button | More Depth Gridlines Options... |
| ChartPrimaryHorizontalAxisOptionsDialog | button | More Primary Horizontal Axis Options... |
| ChartPrimaryVerticalAxisOptionsDialog | button | More Primary Vertical Axis Options... |
| ChartSecondaryHorizontalAxisOption | button | More Secondary Horizontal Axis Options... |
| ChartSecondaryVerticalAxisOptionsDialog | button | More Secondary Vertical Axis Options... |
| ChartDepthAxisOptionsDialog | button | More Depth Axis Options... |
| ChartDataTableOptionsDialog | button | More Data Table Options... |
| ChartTrendlineOptionsDialog | button | More Trendline Options... |
| ChartErrorBarsOptionsDialog | button | More Error Bars Options... |
| ChartUpDownBarsOptionsDialog | button | More Up/Down Bars Options... |
| ChartPlotAreaOptionsDialog | button | More Plot Area Options... |
| ChartWallOptionsDialog | button | More Walls Options... |

| idMso | Control Type | Label |
|-----------------------------------|---------------------|---------------------------------|
| ChartFloorOptionsDialog | button | More Floor Options... |
| ChartSecondaryHorizontalAxisTitle | gallery | Secondary Horizontal Axis Title |
| ChartSecondaryVerticalAxisTitle | gallery | Secondary Vertical Axis Title |
| ChartSecondaryHorizontalGridlines | gallery | Secondary Horizontal Gridlines |
| ChartSecondaryVerticalGridlines | gallery | Secondary Vertical Gridlines |
| ChartSecondaryHorizontalAxis | gallery | Secondary Horizontal Axis |
| ChartSecondaryVerticalAxis | gallery | Secondary Vertical Axis |
| GroupAddInsCustomToolbars | group | Custom Toolbars |
| MoviePlayAutomatically | dropDown | Play Movie: |
| ObjectBringToFrontMenu | splitButton | Bring to Front |
| ObjectSendToBackMenu | splitButton | Send to Back |
| ObjectsGroupMenu | menu | Group |
| ViewDisplayInColor | toggleButton | Color |
| FileSaveAsPowerPoint97_2003 | button | PowerPoint 97-2003 Presentation |
| FileSaveAsMenu | splitButton | Save As Other Format |
| FilePrintMenu | splitButton | Preview and Print |
| FilePermissionRestrictMenu | menu | Restrict Permission |
| DocumentPanelTemplate | button | Document Panel |
| GroupModify | group | Modify |
| BevelShapeGallery | gallery | Bevel |
| _3DBevelPictureTopGallery | gallery | Bevel |
| TableStyleClearTable | button | Clear Table |
| GroupTable | group | Table |
| GroupTableCellSize | group | Cell Size |
| TableEffectsMenu | menu | Effects |
| GlowColorPicker | gallery | More Glow Colors |
| RecolorColorPicker | gallery | More Variations |
| GlowColorMoreColorsDialog | button | More Colors... |
| PictureRecolorMoreColorsDialog | button | More Colors... |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|-------------------------|
| SmartArtAddBullet | button | Add Bullet |
| TableLockAspectRatio | checkBox | Lock Aspect Ratio |
| SlideMasterPicturePlaceholderInsert | button | Picture |
| SlideMasterInsertPlaceholderMenu | splitButton | Insert Placeholder |
| TableBorderColorMoreColorsDialog | button | More Border Colors... |
| TableFillColorMoreColorsDialog | button | More Fill Colors... |
| PictureChange | button | Change Picture... |
| GroupWordArtStyles | group | WordArt Styles |
| TextFillColorPicker | gallery | Text Fill |
| TextOutlineColorPicker | gallery | Text Outline |
| TextOutlineColorMoreColorsDialog | button | More Outline Colors... |
| TextEffectsMenu | menu | Text Effects |
| TextStylesGallery | gallery | Quick Styles |
| WordArtClear | button | Clear WordArt |
| TextPictureFill | button | Picture... |
| TextFillGradientGallery | gallery | Gradient |
| TextFillMoreGradientsDialog | button | More Gradients... |
| TextFillTextureGallery | gallery | Texture |
| TextOutlineDashesGallery | gallery | Dashes |
| TextOutlineMoreLinesDialog | button | More Lines... |
| TextOutlineWeightGallery | gallery | Weight |
| TextEffectShadowGallery | gallery | Shadow |
| TextEffectsMoreShadowsDialog | button | Shadow Options... |
| TextEffectsBevelMore3DOptionsDialog | button | 3-D Options... |
| TextEffects3DRotationGallery | gallery | 3-D Rotation |
| TextEffects3DRotationOptionsDialog | button | 3-D Rotation Options... |
| TextEffectGlowGallery | gallery | Glow |
| TextGlowColorPicker | gallery | More Glow Colors |
| TextGlowColorMoreColorsDialog | button | More Colors... |

| idMso | Control Type | Label |
|-------------------------------------|---------------------|---------------------------------------|
| TextReflectionGallery | gallery | Reflection |
| ShapeEffectsMenu | menu | Shape Effects |
| TransitionSoundLoopUntilNextSound | toggleButton | Loop Until Next Sound |
| GroupChartCurrentSelection | group | Current Selection |
| GroupChartLabels | group | Labels |
| BevelTextGallery | gallery | Bevel |
| PictureCorrectionsDialog | button | Picture Corrections Options... |
| SmartArtAddShapeSplitMenu | splitButton | Add Shape Options |
| MasterNotesPageOrientation | gallery | Notes Page Orientation |
| GroupInkFormat | group | Format |
| InkColorPicker | gallery | Color |
| TextDirectionMoreOptionsDialog | button | More Options... |
| TextAlignMoreOptionsDialog | button | More Options... |
| ParagraphMoreColumnsDialog | button | More Columns... |
| SlideBackgroundReset | button | Reset Slide Background |
| CDAudioPlayTrackAutomatically | dropDown | Play Track: |
| GroupSmartArtShapes | group | Shapes |
| GroupInsertText | group | Text |
| ShapeOutlineColorPicker | gallery | Picture Border |
| GroupColorModeClose | group | Close |
| GroupPageSetupNotesMaster | group | Page Setup |
| GroupPageSetupHandoutMaster | group | Page Setup |
| GroupPlaceholdersNotesMaster | group | Placeholders |
| ReplaceMenu | splitButton | Replace... |
| TableSelectMenuPowerPoint | menu | Select |
| TableDeleteRowsAndColumnsMenu | menu | Delete |
| GroupCDAudioSetup | group | Set Up |
| PrintOptionsMenu | menu | Options |
| ReviewDeleteAllMarkupInPresentation | button | Delete All Markup in the Presentation |

| idMso | Control Type | Label |
|--------------------------------|---------------------|--------------------------|
| PasteMenu | splitButton | Paste |
| GroupPictureStyles | group | Picture Styles |
| PictureStylesGallery | gallery | Quick Styles |
| PictureEffectsShadowGallery | gallery | Shadow |
| PictureEffectsGlowGallery | gallery | Glow |
| PictureEffectsSoftEdgesGallery | gallery | Soft Edges |
| PictureReflectionGallery | gallery | Reflection |
| PictureRotationGallery | gallery | 3-D Rotation |
| InkToolsClose | button | Close Ink Tools |
| GroupViewShowHide | group | Show/Hide |
| GroupWindow | group | Window |
| ViewGridlines | checkBox | View Gridlines |
| GroupPresentationViews | group | Presentation Views |
| GroupColorGrayscale | group | Color/Grayscale |
| FileDocumentEncrypt | toggleButton | Encrypt Document |
| WordArtFormatDialog | button | Format Text Effects... |
| ObjectRotationOptionsDialog | button | More Rotation Options... |
| MoreTextureOptions | button | More Textures... |
| TextFillColorMoreColorsDialog | button | More Fill Colors... |
| ZoomFitToWindow | button | Fit to Window |
| GroupPrintPreviewPageSetup | group | Page Setup |
| ViewDirectionMenu | menu | View Direction |
| FileEmailAsPdfEmailAttachment | button | E-mail as PDF Attachment |
| FileEmailAsXpsEmailAttachment | button | E-mail as XPS Attachment |
| PictureEffectsMenu | menu | Picture Effects |
| PictureShapeGallery | gallery | Change Shape |
| PhotoAlbumInsertMenu | splitButton | Photo Album... |
| GroupChartBackground | group | Background |
| GroupChartAnalysis | group | Analysis |

| idMso | Control Type | Label |
|------------------------------------|---------------------|-------------------------------------|
| MenuPublish | menu | Publish |
| FileCompatibilityCheckerPowerPoint | button | Run Compatibility Checker |
| TableBackgroundPictureFill | button | Picture... |
| UpgradePresentation | button | Convert |
| FileSaveAsOtherFormats | button | Save As |
| FileSaveAsPowerPointPptx | button | PowerPoint Presentation |
| FileSaveAsPowerPointPpsx | button | PowerPoint Show |
| FontDialogPowerPoint | button | Font... |
| PowerPointParagraphDialog | button | Paragraph... |
| PowerPointPageSetup | button | Page Setup... |
| ShapeQuickStylesHome | gallery | Quick Styles |
| GalleryAllShapesAndTextboxes | gallery | Shapes |
| TableTextStylesGallery | gallery | Quick Styles |
| GroupTextStylesTable | group | WordArt Styles |
| ObjectsArrangeMenu | menu | Arrange |
| TextFillMoreTextures | button | More Textures... |
| GroupMacros | group | Macros |
| AdvertisePublishAs | button | Find add-ins for other file formats |
| GroupPermission | group | Protect |
| ReviewProtectPresentationMenu | menu | Protect Presentation |
| AlternativeText | button | Size and Position... |
| FileCheckOutDiscard | button | Discard Check Out |
| MdiChildSystemMenu | menu | Document |

3.2 imageMso Table

| imageMso |
|-----------------|
| Spelling |
| FileSave |
| FilePrint |
| ZoomOnePage |

| |
|----------------------------------|
| imageMso |
| ZoomPageWidth |
| Zoom100 |
| TableInsert |
| ColumnsDialog |
| Numbering |
| Bullets |
| PageOrientationPortraitLandscape |
| OutdentClassic |
| IndentClassic |
| DrawingInsert |
| ChartInsert |
| FileNew |
| Copy |
| Cut |
| Paste |
| FileOpen |
| EnvelopesAndLabelsDialog |
| ZoomPrintPreviewExcel |
| PenComment |
| Folder |
| Repeat |
| UpArrow2 |
| RightArrow2 |
| DownArrow2 |
| LeftArrow2 |
| Clear |
| Breakpoint |
| Piggy |
| Superscript |
| Subscript |

| imageMso |
|----------------------|
| HappyFace |
| UnderlineDouble |
| UnderlineWords |
| FontSizeIncreaseWord |
| FontSizeDecreaseWord |
| _0 |
| _1 |
| _2 |
| _3 |
| _4 |
| _5 |
| _6 |
| _7 |
| _8 |
| _9 |
| A |
| B |
| C |
| D |
| E |
| F |
| G |
| H |
| I |
| J |
| K |
| L |
| M |
| N |
| O |

| imageMso |
|------------------------------|
| P |
| Q |
| R |
| S |
| T |
| U |
| V |
| W |
| X |
| Y |
| Z |
| FileClose |
| TableAutoFormat |
| FormatPainter |
| FilePrintPreview |
| PickUpStyle |
| PasteApplyStyle |
| Bold |
| Italic |
| Underline |
| DarkShading |
| ParagraphMarks |
| AlignLeft |
| AlignRight |
| AlignCenter |
| AlignJustify |
| ContextHelp |
| HeaderFooterPageNumberInsert |
| Undo |
| Redo |

| imageMso |
|-----------------------------|
| ControllLine |
| ControlRectangle |
| OutlinePromote |
| OutlineDemote |
| OutlineMoveUp |
| OutlineMoveDown |
| OutlineDemoteToBodyText |
| OutlineExpand |
| OutlineCollapse |
| TextBoxInsert |
| FileFind |
| FindDialog |
| TableExcelSpreadsheetInsert |
| AutoFormat |
| BorderTop |
| BorderBottom |
| BorderLeft |
| BorderRight |
| BorderInside |
| BorderOutside |
| BorderNone |
| MailMergeGoToFirstRecord |
| MailMergeGoToPreviousRecord |
| MailMergeGoToNextRecord |
| MailMergeGotToLastRecord |
| MailMergeMergeToDocument |
| MailMergeMergeToPrinter |
| MailMergeAutoCheckForErrors |
| DataFormSource |
| MailMergeResultsPreview |

| imageMso |
|---------------------------|
| ObjectsGroup |
| ObjectsUngroup |
| ObjectBringToFront |
| ObjectSendToBack |
| AboveText |
| BehindText |
| ObjectBringForward |
| ObjectSendBackward |
| Magnifier |
| PrintPreviewShrinkOnePage |
| MultiplePages |
| ViewFullScreenView |
| ViewRulerPowerPoint |
| VoiceInsert |
| ObjectsSelect |
| TableFind |
| MacroRecord |
| MacroRecorderPause |
| MacroPlay |
| ObjectFlipHorizontal |
| ObjectFlipVertical |
| ObjectRotateRight90 |
| ObjectRotateLeft90 |
| ShapeFreeform |
| GroupDrawing |
| ObjectEditPoints |
| CalloutOptions |
| SortUp |
| SortDown |
| TableDesign |

| imageMso |
|---------------------------------|
| DataFormAddRecord |
| DataFormDeleteRecord |
| FieldsUpdate |
| DatabaseInsert |
| GridSettings |
| WordPicture |
| FormControlEditBox |
| FormControlCheckBox |
| FormControlComboBox |
| PropertySheet |
| FieldShading |
| ViewDraftView |
| Lock |
| AutoSum |
| MasterDocumentShow |
| MasterDocumentCreateSubdocument |
| MasterDocumentUnlinkSubdocument |
| MasterDocumentInsertSubdocument |
| MasterDocumentSplitSubdocuments |
| MasterDocumentMergeSubdocuments |
| MasterDocumentLockSubdocument |
| HeaderOrFooterShow |
| HeaderFooterPreviousSection |
| HeaderFooterNextSection |
| AlignDialog |
| MailMergeDocument |
| MergeOptions |
| MailMergeHelper |
| PageSetupDialog |
| BodyTextHide |

| imageMso |
|----------------------------|
| HeaderFooterLinkToPrevious |
| OutlineShowFirstLineOnly |
| OutlineShowTextFormatting |
| FontDialog |
| StylesDialogClassic |
| RoutingSlip |
| FootnoteInsert |
| MicrosoftExcel |
| MicrosoftAccess |
| Schedule |
| MicrosoftVisualFoxPro |
| MicrosoftPowerPoint |
| MicrosoftPublisher |
| MicrosoftProject |
| SadFace |
| Pushpin |
| Camera |
| FormControlButton |
| Calculator |
| ViewPrintLayoutView |
| FieldCodes |
| DropCapOptionsDialog |
| Strikethrough |
| TextSmallCaps |
| CellsDelete |
| TableRowsDelete |
| TableColumnsDelete |
| CellsInsertDialog |
| TableRowsInsertWord |
| QueryInsertColumns |

| imageMso |
|----------------------------------|
| WindowsArrangeAll |
| MarginsAdjust |
| ViewGridlinesWord |
| SubdocumentOpen |
| WindowSplit |
| WindowNew |
| LegalBlackline |
| ReviewAcceptOrRejectChangeDialog |
| TextAllCaps |
| PictureDisassemble |
| SymbolInsert |
| ChangeCaseDialogClassic |
| FontSizeDecrease1Point |
| FontSizeIncrease1Point |
| Repaginate |
| ReplaceDialog |
| StartOfLine |
| EndOfLine |
| PagePrevious |
| PageNext |
| TextBoxInsertVertical |
| StartOfDocument |
| EndOfDocument |
| Grammar |
| FileCloseOrCloseAll |
| TextToOrFromTable |
| TableRowsOrColumnsOrCellsInsert |
| TableRowsOrColumnsOrCellsDelete |
| RedoOrRepeat |
| ObjectsRegroup |

| imageMso |
|-----------------------|
| _3DStyle |
| TipWizardHelp |
| AutoFormatChange |
| AddressBook |
| Reply |
| ReplyAll |
| Forward |
| MailMove |
| MailDelete |
| MessagePrevious |
| MessageNext |
| CheckNames |
| MailSelectNames |
| PrintAreaSetPrintArea |
| PasteFormatting |
| FillRight |
| FillDown |
| EqualSign |
| PlusSign |
| MinusSign |
| MultiplicationSign |
| DivisionSign |
| ExponentiationSign |
| ParenthesisLeft |
| ParenthesisRight |
| ColonSign |
| CommaSign |
| PercentSign |
| DollarSign |
| FunctionWizard |

| imageMso |
|------------------------------------|
| AsianLayoutCharacterScaling |
| ConstrainNumeric |
| LightShading |
| AccountingFormat |
| PercentStyle |
| CommaStyle |
| DecimalsIncrease |
| DecimalsDecrease |
| MergeCenter |
| FontSizeIncrease |
| FontSizeDecrease |
| TextOrientationVertical |
| TextOrientationRotateUp |
| TextOrientationRotateDown |
| AlignDistributeHorizontallyClassic |
| ShapeScribble |
| ChartAreaChart |
| Chart3DBarChart |
| Chart3DColumnChart |
| Chart3DPieChart |
| ChartRadarChart |
| OutlineSymbolsShowHide |
| TableSelectVisibleCells |
| SelectCurrentRegion |
| FreezePanels |
| ZoomIn |
| ZoomOut |
| FormControlRadioButton |
| FormControlScrollBar |
| FormControlListBox |

| imageMso |
|------------------------------------|
| TraceDependentRemoveArrows |
| TraceDependents |
| TracePrecedentsRemoveArrows |
| TraceRemoveAllArrows |
| FileUpdate |
| ReadOnly |
| AutoFilterClassic |
| Refresh |
| PivotTableFieldSettings |
| PivotTableShowPages |
| OutlineShowDetail |
| TraceError |
| OutlineHideDetail |
| AlignDistributeVerticallyClassic |
| FormControlGroupBox |
| FormControlSpinner |
| TabOrder |
| RunDialog |
| FormControlCombinationListEdit |
| FormControlCombinationDropDownEdit |
| FormControlLabel |
| Delete |
| Fish |
| Coffee |
| Heart |
| Diamond |
| Spade |
| Club |
| ViewSheetGridlines |
| TracePrecedents |

| imageMso |
|---------------------------|
| Info |
| CodeEdit |
| InsertDialog |
| ApplyFilter |
| DatasheetView |
| SortAndFilterAdvanced |
| ControlSubFormReport |
| FieldList |
| ViewsFormView |
| Grouping |
| AdpPrimaryKey |
| ControlBoundObjectFrame |
| ControlUnboundObjectFrame |
| PageBreakInsertOrRemove |
| PrintSetupDialog |
| CreateFormInDesignView |
| CreateQueryFromWizard |
| CreateReportInDesignView |
| MacroConditions |
| MacroNames |
| ControlToggleButton |
| DatabaseRelationships |
| TableIndexes |
| ViewsAdpDiagramSqlView |
| QueryTableNamesShowHide |
| QueryShowTable |
| QuerySelectQueryType |
| QueryCrosstab |
| QueryMakeTable |
| QueryUpdate |

| |
|----------------------------|
| imageMso |
| QueryAppend |
| QueryDelete |
| QueryParameters |
| RecordsSaveRecord |
| GoToNewRecord |
| RowHeight |
| ColumnWidth |
| RecordsFreezeColumns |
| GridlinesGallery |
| OleObjectctInsert |
| ControlToolboxOutlook |
| SnapToGrid |
| SizeToFit |
| PageHeaderOrFooterShowHide |
| FormHeaderOrFooterShowHide |
| First10RecordsPreview |
| ControlSpecialEffectRaised |
| ControlSpecialEffectSunken |
| AutoDial |
| FindNext |
| PasteDuplicate |
| DatabasePermissions |
| ControlAlignToGrid |
| ControlSpecialEffectFlat |
| CreateTableInDesignView |
| MacroDefault |
| ModuleInsert |
| FilterToggleFilter |
| FilterClearAllFilters |
| Head |

| imageMso |
|------------------------------------|
| ReminderSound |
| CreateModule |
| RelationshipsDirectRelationships |
| RelationshipDesignAllRelationships |
| ControlWizards |
| MergeToWord |
| FilterAdvancedByForm |
| CreateMacro |
| AutoFormatWizard |
| PrintPreviewZoomTwoPages |
| FilterBySelection |
| RecordsDeleteRecord |
| QueryBuilder |
| DatabaseDocumenter |
| DatabaseAnalyzePerformance |
| DatabaseAnalyzeTable |
| ObjectsAlignLeft |
| ObjectsAlignRight |
| ObjectsAlignTop |
| ObjectsAlignBottom |
| ObjectsAlignCenterHorizontal |
| ObjectsAlignMiddleVertical |
| SlideNew |
| ClipArtInsert |
| CreateHandoutsInWord |
| Shadow |
| ObjectRotateFree |
| ShapesMoreShapes |
| CopyToPersonalContacts |
| ParagraphSpacingIncrease |

| imageMso |
|--------------------------|
| ParagraphSpacingDecrease |
| SlideMasterMasterLayout |
| OrganizationChartInsert |
| CollapseAll |
| OutlineExpandAll |
| CombineCharacters |
| DoubleStrikethrough |
| QueryInsertColumn |
| EncryptMessage |
| DigitallySignMessage |
| CreateMailRule |
| ViewNormalViewExcel |
| ViewPageBreakPreviewView |
| SlideHide |
| AnimationCustom |
| PictureCrop |
| SlideShowRehearseTimings |
| ViewSlideView |
| ViewOutlineView |
| ViewSlideSorterView |
| ViewNotesPageView |
| ViewSlideShowView |
| ViewSlideMasterView |
| FileCloseAll |
| FileSaveAs |
| SaveAll |
| AdvancedFileProperties |
| DocumentTemplate |
| FileExit |
| PasteSpecialDialog |

| imageMso |
|----------------------------------|
| SelectAll |
| GoTo |
| BookmarkInsert |
| FileLinksToFiles |
| ViewOnlineLayoutViewClassic |
| HeaderFooterInsert |
| FootnotesEndnotesShow |
| BreakInsertDialog |
| DateAndTimeInsert |
| NumberInsert |
| FieldInsert |
| CaptionInsert |
| CrossReferenceInsert |
| TextFromFileInsert |
| ParagraphDialog |
| BordersShadingDialog |
| TextDirectionOptionsDialog |
| BulletsAndNumberingBulletsDialog |
| AutoFormatDialog |
| SetLanguage |
| WordCount |
| AutoCorrect |
| EnvelopesAndLabels |
| LabelsDialog |
| MergeCells |
| SplitCells |
| TableRowSelect |
| TableColumnSelect |
| TableSelect |
| TableRepeatHeaderRows |

| imageMso |
|-------------------------|
| ConvertTextToTable |
| TableFormulaDialog |
| TableSplitTable |
| ShowClipboard |
| TechnicalSupport |
| ImeDictionaryUpdate |
| OutlookTaskCreate |
| WindowMinimize |
| WindowRestore |
| WindowClose |
| WindowSaveWorkspace |
| SheetDelete |
| ViewFormulaBar |
| SheetInsert |
| FormatCellsDialog |
| DataFormExcel |
| OutlineSubtotals |
| Consolidate |
| WindowHide |
| WindowUnhide |
| FillUp |
| FillLeft |
| ClearFormats |
| NameCreateFromSelection |
| SheetProtect |
| ReviewProtectWorkbook |
| MacroRelativeReferences |
| Filter |
| SortClear |
| AdvancedFilterDialog |

| imageMso |
|---------------------------------|
| OutlineSettings |
| PrintPreviewClose |
| HeaderFooterClose |
| ZoomDialog |
| SortDialogClassic |
| ConvertTableToText |
| PictureInsertFromFilePowerPoint |
| ExchangeFolder |
| VisualBasicReferences |
| ViewCustomViews |
| SheetBackground |
| ChartEditDataSource |
| ChartPlacement |
| CalculateNow |
| ObjectEditDialog |
| ObjectFormatDialog |
| QueryRunQuery |
| ControlImage |
| RulerShowHide |
| GridShowHide |
| ContentsAndIndex |
| Help |
| PivotTableEnableSelection |
| PivotTableListFormulas |
| PivotTableSelectData |
| PivotTableSelectLabelAndData |
| PivotTableSelectLabel |
| CalculateSheet |
| FontColorMoreColorsDialog |
| FillEffects |

| imageMso |
|--|
| TextOrientationAngleCounterclockwise |
| TextOrientationAngleClockwise |
| HyperlinkOpenExcel |
| OpenStartPage |
| WebGoBack |
| WebGoForward |
| AddToFavorites |
| BrowsePrevious |
| BrowseNext |
| BrowseSelector |
| SmartArtInsert |
| ShapeRerouteConnectors |
| ObjectNudgeUp |
| ObjectNudgeDown |
| ObjectNudgeLeft |
| ObjectNudgeRight |
| ShapeCurve |
| ShapeStraightConnector |
| ShapeElbowConnector |
| ObjectFillMoreColorsDialog |
| ObjectBorderOutlineColorMoreColorsDialog |
| LineStylesDialog |
| ArrowsMore |
| TextEffectAlignment |
| TextEffectTracking |
| WordArtVerticalText |
| WordArtEvenTextHeightClassic |
| ContrastMore |
| ContrastLess |
| BrightnessMore |

| imageMso |
|-----------------------------------|
| BrightnessLess |
| ShadowNudgeUpClassic |
| ShadowNudgeDownClassic |
| ShadowNudgeLeftClassic |
| ShadowNudgeRightClassic |
| ObjectShadowColorMoreColorsDialog |
| HighImportance |
| LowImportance |
| AttachMenu |
| InviteAttendees |
| AcceptInvitation |
| DeclineInvitation |
| TentativeAcceptInvitation |
| NewContact |
| NewTask |
| NewAppointment |
| TextAlignLeft |
| TextAlignCenter |
| ShapeRectangle |
| ShapeRoundedRectangle |
| ShapeIsoscelesTriangle |
| ShapeOval |
| ShapeSmileyFace |
| ShapeDonut |
| ShapeLeftBrace |
| ShapeRightBrace |
| ShapeArc |
| ShapeLightningBolt |
| ShapeHeart |
| ShapeRightArrow |

| imageMso |
|--------------------------------|
| ShapeLeftArrow |
| ShapeUpArrow |
| ShapeDownArrow |
| ShapeRoundedRectangularCallout |
| ShapeStar |
| ShapeSeal8 |
| ShapeSeal16 |
| ShapeSeal24 |
| TextAlignRight |
| TextAlignLetterJustify |
| TextAlignWordJustify |
| TextAlignStretchJustify |
| PictureReset |
| PictureRecolorAutomatic |
| PictureRecolorGrayscale |
| PictureRecolorBlackAndWhite |
| PictureRecolorWashout |
| TextWrappingSquare |
| TextWrappingTight |
| TextWrappingNoneClassic |
| TextWrappingEditWrapPoints |
| _3DEffectsOnOffClassic |
| _3DTiltDownClassic |
| _3DTiltUpClassic |
| _3DTiltLeftClassic |
| _3DTiltRightClassic |
| _3DExtrusionDirectionClassic |
| _3DLightingClassic |
| _3DSurfaceMaterialClassic |
| _3DExtrusionDepthNoneClassic |

| imageMso |
|----------------------------------|
| _3DExtrusionDepth36Classic |
| _3DExtrusionDepth72Classic |
| _3DExtrusionDepth144Classic |
| _3DExtrusionDepth288Classic |
| _3DExtrusionDepthInfinityClassic |
| _3DExtrusionPerspectiveClassic |
| _3DExtrusionParallelClassic |
| _3DLightingFlatClassic |
| _3DLightingNormalClassic |
| _3DLightingDimClassic |
| _3DSurfaceMatteClassic |
| _3DSurfacePlasticClassic |
| _3DSurfaceMetalClassic |
| _3DSurfaceWireFrameClassic |
| ObjectEditText |
| SnapToShapes |
| TextWrappingMenuClassic |
| WindowsArrangeIcons |
| PictureFormatDialog |
| ViewVisualBasicCode |
| RemoveFromCalendar |
| MasterViewClose |
| CreateShortcutMenuFromMacro |
| DrawingNewClassic |
| HyperlinkInsert |
| HyperlinkEdit |
| ReviewNewComment |
| ReviewPreviousComment |
| ReviewNextComment |
| ReviewDeleteComment |

| imageMso |
|--|
| ReviewShowOrHideComment |
| ReviewShowAllComments |
| PivotTableOptions |
| DesignMode |
| WordArtInsertDialogClassic |
| FormFieldProperties |
| PhoneticGuideEdit |
| FullScreenViewClassic |
| PhoneticGuideSettings |
| PhoneticGuideFieldShow |
| CircularReferences |
| MasterDocumentExpandOrCollapseSubdocuments |
| Chart3DConeChart |
| InternationalCurrency |
| ObjectsAlignCenterInFormHorizontally |
| ObjectsAlignCenterInFormVertically |
| SizeToControlWidth |
| SizeToControlHeight |
| SizeToControlHeightAndWidth |
| HorizontalSpacingDecrease |
| HorizontalSpacingIncrease |
| ObjectsAlignDistributeHorizontallyRemove |
| VerticalSpacingDecrease |
| VerticalSpacingIncrease |
| ObjectsAlignDistributeVerticallyRemove |
| ObjectsArrangeBottom |
| ObjectsArrangeRight |
| CancelMeeting |
| Private |
| AcceptTask |

| imageMso |
|-----------------------------|
| SaveAndNew |
| CopyFolder |
| EmptyTrash |
| RecordInJournal |
| MarkAsUnread |
| CopyToFolder |
| MoveToFolder |
| ShapeFillColorPickerClassic |
| ControlLineColorPicker |
| VisualBasic |
| DoubleBottomBorder |
| BorderThickBottom |
| BorderTopAndBottom |
| BorderTopAndDoubleBottom |
| BorderTopAndThickBottom |
| BordersAll |
| BorderThickOutside |
| SlideBackgroundFormatDialog |
| AutoSummarize |
| ViewDocumentMap |
| ReviewAcceptChange |
| ReviewRejectChange |
| TableDrawBorderPenStyle |
| Font |
| FontSize |
| ZoomClassic |
| CreateMap |
| MergeCellsAcross |
| FieldChooser |
| MessageHeaderToggle |

| imageMso |
|---------------------------------|
| MeetingRequest |
| NewNote |
| RecurrenceEdit |
| SendStatusReport |
| ImagerScan |
| QueryReturnGallery |
| SelectRecord |
| SelectAllRecords |
| TableTestValidationRules |
| RelationshipsClearLayout |
| SaveAsQuery |
| LoadFromQuery |
| DatasheetColumnRename |
| FileServerLinkTables |
| DatasheetColumnLookup |
| RecordsRefreshRecords |
| RelationshipsEditRelationships |
| RelationshipsHideTable |
| ReplicationRecoverDesignMaster |
| ReplicationResolveConflicts |
| ReplicationCreateReplica |
| ReplicationSynchronizeNow |
| SetDatabasePassword |
| DatabaseUserLevelSecurityWizard |
| DatabaseUserAndGroupAccounts |
| ControlSpecialEffectEtched |
| ControlSpecialEffectShadowed |
| ControlSpecialEffectChiseled |
| WindowsCascade |
| PositionFitToWindow |

| imageMso |
|---------------------------|
| BorderInsideHorizontal |
| BorderInsideVertical |
| BorderDiagonalDown |
| BorderDiagonalUp |
| MagicEightBall |
| TextDirectionLeftToRight |
| TextDirectionRightToLeft |
| ActiveXCheckBox |
| FindDialogExcel |
| ActiveXTextBox |
| ActiveXButton |
| ActiveXRadioButton |
| ActiveXListBox |
| ActiveXComboBox |
| ActiveXToggleButton |
| ActiveXSpinButton |
| ActiveXScrollBar |
| ActiveXLabel |
| ShowBcc |
| ShowFrom |
| DefinePrintStyles |
| TagMarkComplete |
| NewContactFromSameCompany |
| ChooseForm |
| RecoverInviteToMeeting |
| FormPublish |
| SkipOccurrence |
| RightToLeftDocument |
| EditQuery |
| DataRangeProperties |

| imageMso |
|--------------------------------|
| RefreshAll |
| RefreshCancel |
| RefreshStatus |
| ClearAll |
| NewPostInThisFolder |
| ReplyToAttendeesWithMessage |
| SaveAndClose |
| AssignTask |
| Recurrence |
| NewMessageToContact |
| NewTaskForContact |
| NewMeetingWithContact |
| NewMessageToAttendees |
| SendUpdate |
| ReplyToAllAttendeesWithMessage |
| PostReplyToFolder |
| ViewAppointmentInCalendar |
| NewJournalEntry |
| NewMailMessage |
| CancelAcceptTask |
| CancelTaskAssignment |
| CancelDeclineTask |
| NewTaskRequest2 |
| RecurrenceEditSeries |
| DataValidation |
| DataValidationCircleInvalid |
| ReviewShareWorkbook |
| ReviewTrackChanges |
| ReviewHighlightChanges |
| DatabaseQueryNew |

| imageMso |
|--------------------------------------|
| DataValidationClearValidationCircles |
| ReviewEditComment |
| TableDrawTable |
| TableEraser |
| TableCellAlignTop |
| TableCellAlignCenterVertically |
| TableCellAlignBottom |
| TableColumnsDistribute |
| TableRowsDistribute |
| FileCompactAndRepairDatabase |
| DatabaseMakeMdeFile |
| DatabaseEncodeDecode |
| SizeToTallest |
| SizeToShortest |
| SizeToWidest |
| SizeToNarrowest |
| QueryUnionQuery |
| QueryDataDefinition |
| QuerySqlPassThroughQuery |
| ClearGrid |
| ActiveXFrame |
| ActiveXImage |
| ShapeConnectorStyleStraight |
| ShapeConnectorStyleElbow |
| ShapeConnectorStyleCurved |
| WordArtEditTextClassic |
| FilterByResource |
| TableInsertCellsDialog |
| DeleteCells2 |
| TableDeleteRows |

| imageMso |
|-------------------------------------|
| TableDeleteColumns |
| Organizer |
| ShadowOnOrOffClassic |
| MacroRecorderStop |
| FileSendAsAttachment |
| AutoSummaryViewByHighlight |
| MasterDocument |
| ChangeCase |
| ListNumFieldInsert |
| ParagraphSpaceBeforeNone |
| ParagraphSpaceBefore |
| ParagraphSpaceAddOrRemoveBefore |
| PagePreviousWord |
| PageNextWord |
| MailMergeReceipientsUseExistingList |
| FootnoteNextWord |
| EndnoteInsertWord |
| IndexMarkEntry |
| CitationMark |
| IndexInsert |
| TableOfContentsDialog |
| TableOfFiguresInsert |
| TableOfAuthoritiesInsert |
| TextBoxLinkCreate |
| TextBoxLinkBreak |
| TextBoxNextLinked |
| TextBoxPreviousLinked |
| CompareAndCombine |
| PrintOptionsMenuWord |
| PageNumberFormat |

| imageMso |
|------------------------------|
| CancelInvitation |
| ContactWebPage |
| AttachItem |
| SendAgain |
| EditComposePage |
| EditReadPage |
| DesignThisForm |
| FileNewDefault |
| FilePrintQuick |
| WindowsTileVertically |
| WindowsTileHorizontally |
| SpellingAndGrammar |
| CopyToPersonalCalendar |
| CopyToPersonalTaskList |
| Post |
| NewOfficeDocument |
| CreateClassModule |
| ControlTabControl |
| ControlPage |
| ReviewPreviousChangeClassic |
| ReviewNextChangeClassic |
| DialMenu |
| SendDefault |
| MessageProperties |
| PictureInsertFromFile |
| TableDrawBorderPenWeight |
| FormatObject2 |
| TableShowGridlines |
| TableBorderPenColorPicker |
| ShowAutoShapesAndDrawingBars |

| imageMso |
|--|
| ShapeStraightConnectorArrow |
| ShapeElbowConnectorArrow |
| SourceControlAddObjects |
| SourceControlGetLatestVersion |
| SourceControlCheckOut |
| SourceControlCheckIn |
| SourceControlUndoCheckOut |
| SourceControlShareObjects |
| SourceControlShowDifferences |
| SourceControlShowHistory |
| SourceControlRun |
| SourceControlProperties |
| SourceControlCreateDatabaseFromProject |
| SourceControlAddDatabase |
| SourceControlOptions |
| SourceControlRefreshStatus |
| DatabaseMoveToSharePoint |
| NewPostInThisFolder3 |
| SlideDelete |
| ViewHandoutMasterView |
| ViewNotesMasterView |
| SlidesReuseSlides |
| SlidesFromOutline |
| MovieFromFileInsert |
| SoundInsertFromFile |
| FontsReplaceFonts |
| SpeakerNotes |
| BlackAndWhiteAutomatic |
| BlackAndWhiteBlack |
| BlackAndWhiteBlackWithGrayscaleFill |

| imageMso |
|----------------------------------|
| BlackAndWhiteBlackWithWhiteFill |
| BlackAndWhiteDontShow |
| BlackAndWhiteGrayWithWhiteFill |
| BlackAndWhiteGrayscale |
| BlackAndWhiteInverseGrayscale |
| BlackAndWhiteWhite |
| BlackAndWhiteLightGrayscale |
| RecordNarration |
| SlideShowSetUpDialog |
| SummarizeSlide |
| TextWrappingTopAndBottom |
| TextWrappingThrough |
| MoveToFolderMenu |
| MacroRecordOrStop |
| PasteAsHyperlink |
| ParagraphDistributed |
| HyphenationOptions |
| TableRowsOrColumnsDistribute |
| MergeOrSplitCells |
| ReviewJapaneseConsistencyChecker |
| ShowGridlines_HideGridlines |
| AutoSummaryResummarize |
| ViewFooter |
| PictureSetTransparentColor |
| DataRefreshAll |
| BorderTopWord |
| BorderBottomWord |
| BorderLeftWord |
| BorderRightWord |
| TextDirection |

| imageMso |
|---------------------------------|
| SlideShowCustom |
| DuplicateSelectedSlides |
| ActionInsert |
| SlideMiniature |
| PivotTableReport |
| ControlSetControlDefaults |
| ControlActiveX |
| FileNewDatabase |
| FileOpenDatabase |
| FileDatabaseProperties |
| DatasheetColumnDelete |
| SelectAllAccess |
| QueryTotalsShowHide |
| MacroConvertMacrosToVisualBasic |
| ViewsDesignView |
| SlideShowInAWindow |
| HangulHanjaConversionPowerPoint |
| ReturnToTaskList |
| MarkTaskComplete |
| NewMailMessage2 |
| PostReply |
| ShowGridOutlook |
| SizeToGridOutlook |
| ControlSnapToGrid |
| MacroSingleStep |
| MicrosoftOnTheWeb01 |
| MicrosoftOnTheWeb02 |
| MicrosoftOnTheWeb03 |
| MicrosoftOnTheWeb04 |
| MicrosoftOnTheWeb05 |

| imageMso |
|-------------------------------------|
| MicrosoftOnTheWeb06 |
| MicrosoftOnTheWeb07 |
| MicrosoftOnTheWeb08 |
| MicrosoftOnTheWeb09 |
| MicrosoftOnTheWeb10 |
| MicrosoftOnTheWeb11 |
| MicrosoftOnTheWeb12 |
| MicrosoftOnTheWeb13 |
| MicrosoftOnTheWeb14 |
| MicrosoftOnTheWeb15 |
| MicrosoftOnTheWeb16 |
| SendItem |
| SlidesPerPage2Slides |
| SlidesPerPage3Slides |
| SlidesPerPage6Slides |
| SlidesPerPageSlideOutline |
| FontConditionalFormatting |
| ReviewProtectAndShareWorkbook |
| ObjectsAlignLeftOutlook |
| ObjectsAlignRightOutlook |
| ObjectsAlignTopOutlook |
| ObjectsAlignBottomOutlook |
| ObjectsAlignCenterHorizontalOutlook |
| ObjectsAlignMiddleVerticalOutlook |
| MicrosoftOnTheWeb17 |
| SizeToGridAccess |
| SizeToFitAccess |
| FieldsManage |
| HorizontalSpacingMakeEqual |
| VerticalSpacingMakeEqual |

| imageMso |
|-----------------------------------|
| ObjectsAlignToGridOutlook |
| SortAscendingWord |
| SortDescendingWord |
| OutlineGroup |
| OutlineUngroup |
| IndentIncreaseExcel |
| IndentDecreaseExcel |
| ControlAdvancedProperties |
| SaveAttachments |
| ViewVisualBasicCodeAccess |
| FrameInsertHorizontal |
| FormatPictureOrShapeDialogClassic |
| FileSendToPowerPoint |
| DeclineTask |
| AutoFormatNow |
| DataFormWord |
| ReviseContents |
| BulletListDefault |
| NumberListDefault |
| OutlineNumberDefault |
| FormatNumberDefault |
| FootnoteEndnoteDialog |
| TableInsertDialogWord |
| FormFieldClear |
| ObjectBringToFrontOfText |
| ObjectSendBehindText |
| RotateRight2 |
| RotateLeft2 |
| PageBreakInsertWord |
| BordersShadingDialogWord |

| imageMso |
|------------------------------|
| TextBoxWordClassic |
| WordIndent |
| WordOutdent |
| IndentIncreaseWord |
| IndentDecreaseWord |
| SelectObjects |
| Callout |
| ListMacros |
| ReplaceWithAutoText |
| FormatBackground |
| HangulHanjaConversion |
| HeaderSourceEdit |
| IndentIncrease |
| IndentDecrease |
| AsianLayoutFitText |
| AsianLayoutPhoneticGuide |
| AsianLayoutCombineCharacters |
| JapanesePostcardDialog |
| CharacterBorder |
| CharacterShading |
| MoviePlay |
| SlidesPerPage4Slides |
| SlidesPerPage9Slides |
| ViewWebLayoutView |
| PasteAlternative |
| PasteAsNestedTable |
| HyperlinkRemove |
| MacroSecurity |
| HorizontalLineInsert |
| WebPagePreview |

| imageMso |
|-----------------------------|
| TableSelectCell |
| TableDelete |
| TableInsertDialog |
| TableRowsInsertAboveWord |
| TableInsertRowsAbove |
| TableRowsInsertBelowWord |
| TableInsertRowsBelow |
| TableColumnsInsertLeft |
| TableInsertColumnsLeft |
| TableColumnsInsertRight |
| TableInsertColumnsRight |
| TablePropertiesDialog |
| TableOptionsDialog |
| SendCopySendNow |
| SendCopySelectNames |
| SendCopyCheckNames |
| SendCopyOptions |
| SendCopyFlag |
| ConferenceMeetNow |
| SendCopySendToMailRecipient |
| TableOfContentsInFrame |
| ComAddInsDialog |
| FramePropertiesDialog |
| PivotChartInsertClassic |
| PivotFieldListShowHide |
| PictureBulletsInsert |
| FileNewWebPage |
| FileNewEmail |
| ToolboxAudio |
| ToolboxVideo |

| imageMso |
|----------------------------------|
| ToolboxMarquee |
| CreateStoredProcedure |
| FileNewBlankDocument |
| FileNewDialogClassic |
| ViewNormalViewPowerPoint |
| SynchronizeHtml |
| IgnoreHtmlChanges |
| FileSaveAsWebPage |
| HorizontalLineInsertClassic |
| GetExternalDataFromWeb |
| FrameCreateAbove |
| FrameCreateBelow |
| FrameCreateLeft |
| FrameCreateRight |
| FrameDelete |
| RelationshipsReport |
| NewView |
| GetExternalDataFromText |
| ServerConnection |
| RecordsInsertSubdatasheet |
| AdpDiagramNewLabel |
| AdpDiagramAddRelatedTables |
| AdpDiagramShowRelationshipLabels |
| AdpDiagramViewPageBreaks |
| AdpDiagramRecalculatePageBreaks |
| AdpDiagramArrangeSelection |
| AdpDiagramArrangeTables |
| AdpDiagramNewTable |
| AdpDiagramColumnProperties |
| AdpDiagramColumnNames |

| imageMso |
|----------------------------------|
| AdpDiagramKeys |
| AdpDiagramNameOnly |
| AdpDiagramCustomView |
| AdpDiagramDeleteTable |
| AdpDiagramHideTable |
| AdpDiagramAutosizeSelectedTables |
| DiagramDeleteRelationship |
| WindowMoveSplit |
| EastAsianEditingMarks |
| FilePublishAsWebPage |
| TableAutoFitContents |
| TableAutoFitWindow |
| TableAutoFitFixedColumnWidth |
| TableCellAlignTopLeft |
| TableCellAlignTopCenter |
| TableCellAlignTopRight |
| TableCellAlignMiddleLeft |
| TableCellAlignMiddleCenter |
| TableCellAlignMiddleRight |
| TableCellAlignBottomLeft |
| TableCellAlignBottomCenter |
| TableCellAlignBottomRight |
| ServerFilterApply |
| WebControlCheckBox |
| WebControlOptionButton |
| WebControlDropDownBox |
| WebControlListBox |
| WebControlTextBox |
| WebControlTextArea |
| WebControlSubmit |

| imageMso |
|----------------------------------|
| WebControlSubmitWithImage |
| WebControlReset |
| WebControlHidden |
| WebControlPassword |
| TableSetLeftToRight |
| TableSetRightToLeft |
| ChineseTranslationDialog |
| TableInsertMultidiagonalCell |
| AsianLayoutCharactersEnclose |
| ServerFilterByForm |
| EnvelopeChineseDialog |
| AdpVerifySqlSyntax |
| AdpDiagramTableModesMenu |
| CreateDiagram |
| RecordsExpandAllSubdatasheets |
| RecordsCollapseAllSubdatasheets |
| ObjectsMultiSelect |
| DatabaseLinedTableManager |
| TranslateToTraditionalChinese |
| TextWrappingBehindText |
| TextWrappingInFrontOfText |
| WatermarkCustomDialog |
| FrameSaveCurrentAs |
| AdpOutputOperationsTableRemove |
| AdpOutputOperationsGroupBy |
| AdpViewDiagramPane |
| AdpViewGridPane |
| AdpViewSqlPane |
| AdpOutputOperationsAddToOutput |
| AdpOutputOperationsSortAscending |

| imageMso |
|-------------------------------------|
| AdpOutputOperationsSortDescending |
| DatabaseAccessBackEnd |
| DatabasePartialReplica |
| TextDirectionContext |
| TranslateToSimplifiedChinese |
| ChineseTranslationMenu |
| FileBackUpSqlDatabase |
| ServerRestoreSqlDatabase |
| FileDropSqlDatabase |
| DatabaseSetLogonSecurity |
| DatabaseSqlServer |
| DatabaseSwitchboardManager |
| WebServerDiscussions |
| FontColorMoreColorsDialogPowerPoint |
| HyperlinksVerify |
| BlackAndWhite |
| CancelRequest |
| ViewMasterDocumentViewClassic |
| HyperlinkInsertPowerPoint |
| AddOrRemoveAttendees |
| NewDistributionList |
| CalculateFull |
| SizeMenu |
| ScriptDebugger |
| RunThisForm |
| MessageFormatPlainText |
| MessageFormatHtml |
| MessageFormatRichText |
| ExportToVCardFile |
| MessageOptions |

| imageMso |
|-----------------------------------|
| MapContactAddress |
| SignatureInsertMenu |
| DesignAFormOutlook |
| WatchWindow |
| FormulaEvaluate |
| LineSpacing |
| WordCountList |
| StylesModifyStyle |
| MacroRun |
| ServerProperties |
| SignatureShow |
| StylesPane |
| SpeechMicrophone |
| SubformInNewWindow |
| NumberingRestart |
| DrawingCanvasInsert |
| ViewsPivotTableView |
| ViewsPivotChartView |
| DiagramRadialInsertClassic |
| DiagramCycleInsertClassic |
| DiagramPyramidInsertClassic |
| DiagramTargetInsertClassic |
| DiagramVennDiagramInsertClassic |
| DiagramChangeToRadialClassic |
| DiagramChangeToCycleClassic |
| DiagramChangeToTargetClassic |
| DiagramChangeToVennDiagramClassic |
| PasteByAppendingTable |
| OrganizationChartInsertAssistant |
| OrganizationChartInsertCoworker |

| imageMso |
|------------------------------------|
| OrganizationChartInsertSubordinate |
| AdpStoredProcedureEditSql |
| DrawingCanvasFit |
| DrawingCanvasResize |
| ColorGrayscaleMenu |
| ViewBackToColorView |
| ViewDisplayInGrayscale |
| ViewDisplayInHighContrast |
| ViewDisplayInPureBlackAndWhite |
| PivotAutoFilter |
| PivotSubtotal |
| PivotRefresh |
| PivotMoveToFieldArea |
| PivotMoveToColumnArea |
| PivotMoveToFilterArea |
| PivotMoveToDetailArea |
| PivotExpandField |
| PivotExportToExcel |
| PivotExpandIndicators |
| PivotDropAreas |
| SearchUI |
| MailMergeMergeToEmail |
| MailMergeMergeToFax |
| MailMergeCreateList |
| DrawingCanvasExpand |
| TextWrappingInLineWithText |
| SlidesPerPage1Slide |
| ReviewSendForReview |
| WorkgroupAdmin |
| AdpDiagramAddTable |

| imageMso |
|--------------------------------------|
| WebComponent |
| PasteSpecial |
| AttachFile |
| DiagramChangeToPyramidClassic |
| DiagramShapeMoveBackwardClassic |
| DiagramShapeMoveForwardClassic |
| SlideMasterPreserveMaster |
| SlideMasterRenameMaster |
| CharacterCodeToggle |
| LinkBarCustom |
| BorderDrawMenu |
| BorderErase |
| BorderStyle |
| StylesStyleVisibility |
| ShowFormulas |
| DatabaseCopyDatabaseFile |
| FileServerTransferDatabase |
| DiagramStylesClassic |
| MailMergeHighlightMergeFields |
| MailMergeWizard |
| OrganizationChartAutoLayout |
| OrganizationChartSelectLevel |
| OrganizationChartSelectBranch |
| OrganizationChartSelectAllAssistants |
| OrganizationChartSelectAllConnectors |
| PivotTableOlapPropertyFields |
| RevealFormatting |
| DiagramReverseClassic |
| DiagramAutoLayoutClassic |
| PasteTableFromExcel |

| imageMso |
|-------------------------------------|
| TranslationPane |
| GoToTableOfContents |
| TableOfContentsUpdate |
| OutlineLevelGallery |
| OutlineShowLevel |
| ErrorChecking |
| BulletsAndNumberingNumberingDialog |
| NumberingContinue |
| FileCheckOut |
| FileCheckIn |
| OrganizationChartLayoutStandard |
| OrganizationChartLayoutBothHanging |
| OrganizationChartLayoutLeftHanging |
| OrganizationChartLayoutRightHanging |
| PivotTableGenerateGetPivotData |
| ReviewReplyWithChanges |
| StylesStyleSeparator |
| BorderDrawLine |
| BorderDrawGrid |
| OutlinePromoteToHeading |
| MicrosoftOutlook |
| ReviewPreviousChange |
| ReviewNextChange |
| ReviewReviewingPane |
| ReviewAcceptChangeClassic |
| ReviewRejectChangeOrDeleteComment |
| PageOrientationLandscape |
| PageOrientationPortrait |
| GetExternalDataImportClassic |
| SlideReset |

| imageMso |
|---------------------------------|
| AnimationOnClick |
| StartAfterPrevious |
| MailMergeMatchFields |
| MailMergeAddressBlockInsert |
| MailMergeGreetingLineInsert |
| MailMergeMergeFieldInsert |
| MailMergeRecipientsEditList |
| PicturesCompress |
| TableAutoFormatStyle |
| PivotAutoCalcMenu |
| PivotChartType |
| PivotChartMultiplePlots |
| PivotChartMultipleUnified |
| PivotSwitchRowColumn |
| DrillInto |
| MailMergeFindRecipient |
| VerticallyDistributed |
| ReviewDeleteCommentPowerPoint |
| ReviewNextCommentPowerPoint |
| ReviewPreviousCommentPowerPoint |
| ReviewShowOrHideMarkup |
| FontSchemes |
| FormFieldReset |
| MailMergeUpdateLabels |
| PivotCollapseFieldAccess |
| PivotShowDetails |
| PivotHideDetails |
| PivotChartLegendShowHide |
| AnimationAudio |
| DiagramFitToContentsClassic |

| imageMso |
|-------------------------------------|
| DiagramResizeClassic |
| DiagramExpandClassic |
| PivotShowAsMenu |
| OrganizationChartResize |
| AcceptProposal |
| AppointmentColor0 |
| AppointmentColor1 |
| AppointmentColor2 |
| AppointmentColor3 |
| AppointmentColor4 |
| AppointmentColor5 |
| AppointmentColor6 |
| AppointmentColor7 |
| AppointmentColor8 |
| AppointmentColor9 |
| AppointmentColor10 |
| AppointmentColorDialog |
| ProposeNewTime |
| ViewAllProposals |
| ReviewDeleteAllMarkupOnSlide |
| PasteOption |
| MailMergeSetDocumentType |
| DiagramShapeInsertClassic |
| OrganizationChartStyle |
| AdpStoredProcedureQuerySelect |
| AdpStoredProcedureQueryMakeTable |
| AdpStoredProcedureQueryUpdate |
| AdpStoredProcedureQueryAppend |
| AdpStoredProcedureQueryAppendValues |
| AdpStoredProcedureQueryDelete |

| imageMso |
|------------------------------|
| ReviewDisplayForReview |
| ReviewNewCommentMenu |
| ReviewAllowUsersToEditRanges |
| SpeakCells |
| SpeakStop |
| SpeakByRows |
| SpeakByColumns |
| SpeakOnEnter |
| Translate |
| PivotClearCustomOrdering |
| PivotFilterBySelection |
| PivotRemoveField |
| PivotGroupItems |
| PivotUngroupItems |
| DrillOut |
| AdpManageIndexes |
| ViewGridlinesPowerPoint |
| AdpDiagramIndexesKeys |
| AdpDiagramRelationships |
| AdpConstraints |
| PhotoAlbumInsert |
| PhotoAlbumEdit |
| DrawingCanvasScale |
| OrgChartScale |
| DiagramScale |
| ProtectDocument |
| BrightnessAndContrastEdit |
| TableInsertExcel |
| ReadingMode |
| ReadingViewClose |

| imageMso |
|-----------------------------|
| PersonaStatusOnline |
| PersonaStatusOffline |
| PersonaStatusAway |
| PersonaStatusBusy |
| GoToMail |
| ResearchPane |
| TableStyleTotalsRow |
| TableRowsInsertAboveExcel |
| TableRowsDeleteExcel |
| TableConvertToRange |
| PrintListRange |
| ReadingModeMini |
| DocumentMapReadingView |
| ReadingViewResearchPane |
| ReadingViewFontSizeIncrease |
| ReadingViewFontSizeDecrease |
| ReadingViewShowPrintedPage |
| ViewRulerWord |
| FileInternetFax |
| XmlExport |
| XmlImport |
| ViewDocumentActionsPane |
| DatabaseObjectDependencies |
| ReviewShowInk |
| InkEraseMode |
| TableColumnsInsertLeftExcel |
| TableColumnsDeleteExcel |
| NewAlert |
| InkDeleteAllInk |
| TableUnlinkExternalData |

| imageMso |
|--------------------------------------|
| TableExportTableToSharePointList |
| LookUp |
| FilePackageForCD |
| DesignXml |
| WindowSideBySide |
| ViewGridlinesFrontPage |
| ChangeBinding |
| ListSynchronize |
| ChangesDiscardAndRefresh |
| TableOpenInBrowser |
| TableResize |
| XmlExpansionPacksExcel |
| FileVersionHistory |
| FileVersionHistoryWord |
| FrontPageToggleBookmark |
| XmlDataRefresh |
| XmlMapProperties |
| WindowSideBySideSynchronousScrolling |
| WindowResetPosition |
| InkColorMoreColorsDialog |
| XmlTransformation |
| XmlEditQuery |
| JotInkStyle1 |
| InsertDrawingCanvas |
| FilePermissionView |
| FilePermission |
| ReadingViewAllowMultiplePages |
| VoiceInsertInComment |
| RmsInvokeBrowser |
| RmsImportanceCheck |

| imageMso |
|---------------------------|
| RmsImportanceUncheck |
| ViewThumbnails |
| Thesaurus |
| InkingStart |
| VoteMenu |
| RmsNavigationBar |
| RmsSendBizcard |
| RmsSendBizcardDesign |
| FindText |
| AudioNoteDelete |
| ViewFullScreenReadingView |
| VisibilityVisible |
| VisibilityHidden |
| VisibilityInherit |
| InkDrawingAndWriting |
| ReviewInkCommentPen |
| ReviewInkCommentEraser |
| NewInternetFax |
| MeetingsWorkspace |
| InkEraser |
| InkStopInkingReadingView |
| StarUnratedEmpty |
| StarUnratedFull |
| StarRatedEmpty |
| StarRatedFull |
| StarRatedHalf |
| OutlookGlobe |
| OutlookGears |
| MsnLogo |
| GoLeftToRight |

| imageMso |
|----------------------------------|
| GoLtrHover |
| GoLtrFocus |
| GoLtrDown |
| GoRtl |
| GoRtlHover |
| GoRtlFocus |
| GoRtlDown |
| StopLeftToRight |
| StopLtrHover |
| StopLtrFocus |
| StopLtrDown |
| StopRtl |
| StopRtlHover |
| StopRtlFocus |
| StopRtlDown |
| ForwardToMobile |
| JunkEmailAddToBlockedSendersList |
| JunkEmailMarkAsNotJunk |
| JunkEmailOptions |
| CategoryCollapse |
| DataSourceCatalogServerScript |
| PermissionRestrict |
| Risks |
| SetPertWeights |
| PanAndZoomWindow |
| FileBackupDatabase |
| OrganizationOutlook |
| ThemeBrowseForThemesPowerPoint |
| AccessNavigationOptions |
| WorkflowComplete |

| imageMso |
|---------------------------|
| WorkflowPending |
| Connections |
| ViewGoBack |
| ViewGoForward |
| ListSetNumberingValue |
| SlideMasterInsertLayout |
| FileViewDigitalSignatures |
| SignaturesLoading |
| AppointmentAttachment |
| ViewsReportView |
| FileCloseDatabase |
| OpenInfopathForm |
| FileWorkflowTasks |
| FileStartWorkflow |
| SharingRequestAllow |
| SharingRequestDeny |
| SharingOpenCalendarFolder |
| SignatureLineInsert |
| SlideThemesGallery |
| ApplicationOptionsDialog |
| BibliographyInsert |
| BibliographyStyle |
| CitationInsert |
| BibliographyManageSources |
| BibliographyAddNewSource |
| LabelInsert |
| BarcodeInsert |
| CongratulatoryEvent |
| CondolatoryEvent |
| ViewPageLayoutView |

| imageMso |
|-------------------------------|
| PivotClearAll |
| ChartStylesGallery |
| ChartLayoutGallery |
| ChartSaveTemplates |
| ChartAxisTitles |
| ChartAxes |
| ChartGridlines |
| ChartMoreElement |
| ChartFormatSelection |
| ChartElementSelector |
| PageMarginsGallery |
| LimeCatsAndDogsGallery |
| DropCapInsertGallery |
| TrustCenter |
| ReadingViewAllowTyping |
| ReadingViewMarginSettingsMenu |
| CalendarInsert |
| SlideShowFromCurrent |
| BulletsGallery |
| NumberingGallery |
| LineSpacingGalleryPowerPoint |
| SlideNewGallery |
| SlideLayoutGallery |
| ShapesInsertGallery |
| ShapeChangeShapeGallery |
| ShapeFillTextureGallery |
| ShapeStylesGallery |
| PageOrientationGallery |
| FileServerTasksMenu |
| FileSendMenu |

| imageMso |
|---------------------------------|
| GroupFont |
| GroupParagraph |
| GroupStyles |
| GroupProofing |
| GroupInsertPages |
| GroupInsertIllustrations |
| GroupShapes |
| GroupWordArtText |
| GroupPageSetup |
| GroupParagraphLayout |
| GroupCitationsAndBibliography |
| GroupFootnotes |
| GroupTableOfContents |
| GroupMailMergeWriteInsertFields |
| GroupMailMergePreviewResults |
| GroupMailMergeFinish |
| GroupChangesTracking |
| GroupComments |
| GroupChanges |
| GroupCompare |
| GroupPictureSize |
| GroupDrawBorders |
| GroupTableAlignment |
| GroupTableProperties |
| GroupTableTools |
| GroupSlides |
| GroupTextBoxText |
| GroupArrange |
| GroupInsertMediaClips |
| GroupSlideThemes |

| imageMso |
|-----------------------------------|
| GroupThemes |
| GroupBackground |
| GroupPreview |
| GroupAnimations |
| GroupTransitionToThisSlide |
| GroupSlideShowStart |
| GroupSlideShowSetup |
| GroupShapeStyles |
| GroupFill |
| GroupLines |
| GroupEffects |
| GroupChartLayouts |
| GroupChartStyles |
| GroupChartAxes |
| GroupChartShapes |
| GroupNumber |
| GroupAlignmentExcel |
| GroupCells |
| GroupSortFilter |
| GroupInsertTablesExcel |
| GroupOrganizationChartShapeInsert |
| GroupPageLayoutScaleToFit |
| GroupPageLayoutSheetOptions |
| GroupFunctionLibrary |
| GroupNamedCells |
| GroupFormulaAuditing |
| GroupGetExternalData |
| GroupConnections |
| GroupOutline |
| GroupDataTools |

| imageMso |
|--------------------------------|
| GroupChangesExcel |
| FollowUpReadMenu |
| StylesManageStyles |
| StylesStyleInspector |
| ObjectEffectPresetGallery |
| PictureEffectsPresetGallery |
| _3DRotationGallery |
| ControlLayoutTabular |
| ControlLayoutStacked |
| ControlLayoutRemove |
| ShapeFillColorPicker |
| OutlineColorPicker |
| ResultsPaneStartFindAndReplace |
| FileDocumentInspect |
| ImportSavedImports |
| ImportAccess |
| ImportExcel |
| ImportTextFile |
| ImportSharePointList |
| ImportXmlFile |
| ImportOdbcDatabase |
| ImportHtmlDocument |
| ImportOutlook |
| ImportDBase |
| ImportParadox |
| ImportLotus |
| ExportSavedExports |
| ExportExcel |
| ExportSharePointList |
| ExportWord |

| imageMso |
|--|
| ExportAccess |
| ExportTextFile |
| ExportXmlFile |
| ExportOdbcDatabase |
| ExportSnapshot |
| ExportHtmlDocument |
| ExportDBase |
| ExportParadox |
| ExportLotus |
| QuickStylesGallery |
| QuickStylesSets |
| ClearFormatting |
| FormRegionOpen |
| FormRegionSave |
| PanningHand |
| BulletsGalleryWord |
| NumberingGalleryWord |
| SharePointListsWorkOffline |
| SynchronizeData |
| SharePointListsDiscardAllChangesAndRefresh |
| FileManageMenu |
| ViewsModeMenu |
| GroupSortAndFilter |
| SortSelectionMenu |
| FiltersMenu |
| RecordsMoreRecordsMenu |
| FieldsMenu |
| RecordsSubdatasheetMenu |
| GroupAutoFormatAccess |
| GroupTextFormatting |

| imageMso |
|----------------------------|
| GroupDataTypeAndFormatting |
| AlignLeftToRightMenu |
| GroupCreateTables |
| GroupCreateForms |
| GroupCreateReports |
| GroupCreateOther |
| GroupSharepointLists |
| GroupCollectData |
| GroupImport |
| ImportMoreMenu |
| GroupExport |
| ExportMoreMenu |
| GroupDatabaseTools |
| DatabasePermissionsMenu |
| GroupMacro |
| GroupDatabaseSourceControl |
| GroupSourceControlShow |
| GroupSourceControlManage |
| GroupRelationships |
| GroupMacroRows |
| GroupQuerySetup |
| GroupQueryType |
| SqlSpecificMenu2 |
| GroupQueryTools |
| GroupControls |
| GroupPageLayoutAccess |
| GroupZoom |
| GroupToolsAccess |
| GroupRichText |
| GroupSizeAndPosition |

| imageMso |
|--------------------------------|
| PivotShowTopAndBottomItemsMenu |
| PivotFormulasMenu |
| FilePublishSlides |
| SlideShowUsePresenterView |
| ArrowStyleGallery |
| OutlineDashesGallery |
| OutlineWeightGallery |
| GroupAlignment |
| GroupPictureTools |
| GroupSize |
| FormatCellsNumberDialog |
| FormatCellsFontDialog |
| CellAlignmentOptions |
| PageSetupPageDialog |
| PageSetupSheetDialog |
| PivotTableLayoutGrandTotals |
| GroupPivotTableActiveField |
| GroupPivotTableLayout |
| GroupPivotTableSort |
| GroupPivotTableShowHide |
| PivotTableLayoutSubtotals |
| GroupPivotTableGroup |
| GroupPivotTableTools |
| GroupPivotTableData |
| GroupPivotTableOptions |
| GroupPivotTableStyles |
| GroupPivotTableStyleOptions |
| WrapText |
| ClearMenu |
| ProofingToolsFlyoutAnchor |

| imageMso |
|---------------------------------|
| PictureInsertMenu |
| ReviewTrackChangesMenu |
| ReviewAcceptChangeMenu |
| ReviewRejectChangeMenu |
| ReviewBalloonsMenu |
| GroupTableStylesPowerPoint |
| GroupTableRowsAndColumns |
| GroupTableData |
| ObjectAlignMenu |
| MovieInsert |
| ObjectRotateGallery |
| FillMenu |
| SelectMenu |
| OrientationMenu |
| MergeCenterMenu |
| AutoSumMenu |
| PrintAreaMenu |
| PageBreakMenu |
| NameDefineMenu |
| RefreshMenu |
| WhatIfAnalysisMenu |
| PivotTableFormulasMenu |
| PivotTableOlapTools |
| PivotTableOptionsMenu |
| AnimationGallery |
| AnimationTransitionGallery |
| AnimationTransitionSpeedGallery |
| ErrorCheckingMenu |
| TraceRemoveArrowsMenu |
| SortFilterMenu |

| imageMso |
|---------------------------------|
| FontColorPicker |
| TableColumnsGallery |
| CellFillColorPicker |
| BorderDoubleBottom |
| ChartTitle |
| ChartPrimaryHorizontalAxisTitle |
| ChartPrimaryVerticalAxisTitle |
| ChartDepthAxisTitle |
| ChartLegend |
| ChartDataLabel |
| ChartPrimaryHorizontalGridlines |
| ChartPrimaryVerticalGridlines |
| ChartDepthGridlines |
| ChartPrimaryHorizontalAxis |
| ChartPrimaryVerticalAxis |
| ChartDepthAxis |
| ChartDataTable |
| ChartTrendline |
| ChartErrorBars |
| ChartLines |
| ChartUpDownBars |
| ChartPlotArea |
| ChartWall |
| ChartFloor |
| GroupMarginsAndPadding |
| ControlLineTypeGallery |
| ControlLineThicknessGallery |
| GroupMacroTools |
| ReplicationOptionsMenu |
| GroupRecords |

| imageMso |
|------------------------------|
| LassoSelect |
| SmartArtAddShape |
| SmartArtIncreaseFontSize |
| SmartArtDecreaseFontSize |
| SmartArtLargerShape |
| SmartArtSmallerShape |
| SmartArtResetGraphic |
| SmartArtResetShape |
| SmartArtTextPane |
| SmartArtEditIn2D |
| SmartArtLayoutGallery |
| SmartArtMoreLayoutsDialog |
| SmartArtStylesGallery |
| SmartArtChangeColorsGallery |
| ObjectEffectSoftEdgesGallery |
| ObjectEffectGlowGallery |
| GradientGallery |
| ObjectEffectShadowGallery |
| WordArtInsertGallery |
| TextEffectTransformGallery |
| TextNoTransform |
| TextPathArchUp |
| TextPathArchDown |
| TextPathCircle |
| TextPathButton |
| TextPlain |
| TextStop |
| TextTriangle |
| TextTriangleInverted |
| TextChevron |

| imageMso |
|---------------------------|
| TextChevronInverted |
| TextRingInside |
| TextRingOutside |
| TextArchUpPour |
| TextArchDownPour |
| TextCirclePour |
| TextButtonPour |
| TextCurveUp |
| TextCurveDown |
| TextCascadeUp |
| TextCascadeDown |
| TextWave1 |
| TextWave2 |
| TextWave3 |
| TextWave4 |
| TextInflate |
| TextDeflate |
| TextInflateBottom |
| TextDeflateBottom |
| TextInflateTop |
| TextDeflateTop |
| TextDeflateInflate |
| TextDeflateInflateDeflate |
| TextFadeRight |
| TextFadeLeft |
| TextFadeUp |
| TextFadeDown |
| TextSlantUp |
| TextSlantDown |
| TextCanUp |

| imageMso |
|--|
| TextCanDown |
| GroupHeaderFooterExcel |
| HeaderFooterHeaderGallery |
| HeaderFooterFooterGallery |
| GroupHeaderFooterElements |
| HeaderFooterPageNumberInsertExcel |
| HeaderFooterNumberOfPagesInsert |
| HeaderFooterCurrentDate |
| HeaderFooterCurrentTimeInsert |
| HeaderFooterFilePathInsert |
| HeaderFooterFileNameInsert |
| HeaderFooterSheetNameInsert |
| HeaderFooterPictureInsert |
| HeaderFooterFormatPicture |
| GroupHeaderFooterOptions |
| FontShadingColorMoreColorsDialog |
| FontColorMoreColorsDialogExcel |
| BorderMoreColorsDialog |
| SheetTabColorMoreColorsDialog |
| PivotTableNewStyle |
| PivotPlusMinusFieldHeadersShowHide |
| PivotTableExpandField |
| PivotCollapseField |
| ConditionalFormattingDataBarsGallery |
| ConditionalFormattingColorScalesGallery |
| ConditionalFormattingIconSetsGallery |
| TableColumnsInsertRightExcel |
| TableRowsInsertBelowExcel |
| ConditionalFormattingHighlightBetween |
| ConditionalFormattingHighlightCompareColumns |

| imageMso |
|---|
| PivotTableStylesGallery |
| FormatAsTableGallery |
| TableStylesGalleryExcel |
| ConditionalFormattingsManage |
| ConditionalFormattingHighlightGreaterThan |
| ConditionalFormattingHighlightLessThan |
| ConditionalFormattingHighlightEqualTo |
| ConditionalFormattingHighlightTextContaining |
| ConditionalFormattingHighlightDateOccuring |
| ConditionalFormattingHighlightDuplicateValues |
| ConditionalFormattingTopNItems |
| ConditionalFormattingTopNPercent |
| ConditionalFormattingBottomNItems |
| ConditionalFormattingBottomNPercent |
| ConditionalFormattingAboveAverage |
| ConditionalFormattingBelowAverage |
| RemoveDuplicates |
| FilterReapply |
| CreateEmail |
| ManageReplies |
| ReviewPreviousCommentWord |
| ReviewNextCommentWord |
| ThemeColorsGallery |
| PivotTableInsert |
| PivotChartInsert |
| PivotTableMove |
| PivotTableChangeDataSource |
| BuildingBlocksCreateNewFromSel |
| HeaderInsertGallery |
| FooterInsertGallery |

| imageMso |
|---|
| CoverPageInsertGallery |
| PageNumberFieldInsertGallery |
| WatermarkGallery |
| EquationInsertGallery |
| QuickTablesInsertGallery |
| QuickPartsInsertGallery |
| SharePointListsDiscardAllChanges |
| SortRemoveAllSorts |
| GroupSmartArtLayouts |
| GroupSmartArtQuickStyles |
| GroupSmartArtChooseColor |
| GroupSmartArtCreateGraphic |
| GroupSmartArtReset |
| GroupSmartArtSize |
| ConditionalFormattingHighlightCellsMenu |
| ConditionalFormattingTopBottomMenu |
| SaveSelectionToQuickPartGallery |
| SaveSelectionToCoverPageGallery |
| SaveSelectionToEquationGallery |
| SaveSelectionToFooterGallery |
| SaveSelectionToHeaderGallery |
| SaveSelectionToPageNumberGallery |
| SaveSelectionToQuickTablesGallery |
| SaveSelectionToWaterMarkGallery |
| MenuView2 |
| FormatCellsMenu |
| ConditionalFormattingClearMenu |
| FontColorCycle |
| CellsInsertSmart |
| CellsDeleteSmart |

| imageMso |
|--|
| ReviewShowMarkupMenu |
| ObjectEditShapeMenu |
| PivotTableLayoutReportLayout |
| PivotTableLayoutShowInCompactForm |
| PivotTableLayoutShowInOutlineForm |
| PivotTableLayoutShowInTabularForm |
| SymbolInsertGallery |
| ControlChart |
| SpellingAccess |
| ViewsDatasheetView |
| PivotTableClearMenu |
| MacroArguments |
| MacroShowAllActions |
| GroupControlsAccess |
| CreateTable |
| CreateTableTemplatesGallery |
| AccessTableContacts |
| AccessTableTasks |
| AccessTableIssues |
| AccessTableEvents |
| AccessTableAssets |
| CreateTableUsingSharePointListsGallery |
| AccessListContacts |
| AccessListTasks |
| AccessListIssues |
| AccessListEvents |
| AccessListAssets |
| AccessListCustom |
| AccessListCustomDatasheet |
| CreateForm |

| imageMso |
|--|
| CreateFormSplitForm |
| CreateFormWithMultipleItems |
| CreateFormMoreFormsGallery |
| AccessFormWizard |
| CreateFormPivotChart |
| AccessFormPivotTable |
| AccessFormDatasheet |
| AccessFormModalDialog |
| CreateFormBlankForm |
| CreateReport |
| AccessReportMore |
| CreateReportFromWizard |
| CreateLabels |
| CreateReportBlankReport |
| CreateOtherObjectsMenu |
| ControlTitle |
| ControlLogo |
| ShowMargins |
| SlideMasterContentPlaceholderInsert |
| SlideMasterTextPlaceholderInsert |
| SlideMasterChartPlaceholderInsert |
| SlideMasterTablePlaceholderInsert |
| SlideMasterDiagramPlaceholderInsert |
| SlideMasterMediaPlaceholderInsert |
| SlideMasterClipArtPlaceholderInsert |
| SlideMasterVerticalTextPlaceholderInsert |
| SlideMasterShowFooters |
| CellStylesGallery |
| CellStyleNew |
| CellStylesMerge |

| imageMso |
|------------------------------------|
| TableStylesGallery |
| TableStyleNew |
| TableStyleClear |
| FilePublishExcelServices |
| PivotTableOlapConvertToFormulas |
| PivotTableLayoutBlankRows |
| TableStyleFirstColumn |
| TableStyleLastColumn |
| TableStyleBandedRows |
| TableStyleBandedColumns |
| TableStyleRowHeaders |
| TableStyleColumnHeaders |
| TableSummarizeWithPivot |
| DisplayRuler |
| GetExternalDataFromAccess |
| GetExternalDataFromOtherSources |
| GetExternalDataExistingConnections |
| GroupThemesExcel |
| PivotPlusMinusButtonsShowHide |
| FileSaveAsPdfOrXps |
| FormattingDataType |
| FormattingRequiredField |
| FormattingFormat |
| ApplyCurrencyFormat |
| ApplyPercentageFormat |
| ApplyCommaFormat |
| FormattingIncreaseDecimals |
| FormattingDecreaseDecimals |
| BusinessCardInsertMenu |
| RecentFileList |

| imageMso |
|---------------------------|
| DataGraphicIconSet |
| SearchLibraries |
| MoreControlsDialog |
| GroupCode |
| GroupRestrictions |
| GroupXml |
| PageScaleToFitWidth |
| PageScaleToFitHeight |
| SlideTransitionApplyToAll |
| AnimationPreview |
| XmlStructure |
| XmlSchema |
| XmlExpansionPacksWord |
| GroupCaptions |
| GroupIndex |
| GroupTableOfAuthorities |
| GroupEditing |
| SelectMenuExcel |
| PrintTitles |
| NameUseInFormula |
| CalculationOptionsMenu |
| XmlSource |
| GroupClipboard |
| GroupInsertTables |
| GroupInsertLinks |
| GroupInsertSymbols |
| GroupInsertBarcode |
| GroupCalculation |
| BordersGallery |
| BordersMoreDialog |

| imageMso |
|--|
| PageScaleToFitOptionsDialog |
| PageSizeGallery |
| AnimationTransitionSoundGallery |
| ObjectPictureFill |
| TextWrappingMenu |
| RecordsAddFromOutlook |
| RecordsSaveAsOutlookContact |
| DatasheetNewField |
| PivotTableGroupSelection |
| PivotTableGroupField |
| SlideShowFromBeginning |
| WindowSwitchWindowsMenuWord |
| WindowSwitchWindowsMenuPowerPoint |
| WindowSwitchWindowsMenuExcel |
| ShapeFillMoreGradientsDialog |
| ShadowOptionsDialog |
| BuildingBlocksOrganizer |
| TableSharePointListsModifyColumnsAndSettings |
| TableListAlertMe |
| TableSharePointListsModifyWorkflow |
| TableListPermissions |
| TableSharePointListsRefreshList |
| CreateQueryInDesignView |
| ControlAttachment |
| GroupFieldsAndColumns |
| DataOptionsMenu |
| FormattingUnique |
| RecordsRefreshMenu |
| GroupLayoutShowHide |
| GroupQueryShowHide |

| imageMso |
|-----------------------------------|
| GroupMacroShowHide |
| GroupPosition |
| GroupAnalyze |
| GroupControlLayout |
| GroupRelationshipsTools |
| GroupTableDesignTools |
| GroupTableRows |
| GroupSharePointList |
| GroupPivotTableFilterAndSort |
| GroupPivotTableToolsAccess |
| GroupPivotTableShowHideAccess |
| GroupPivotTableSelections |
| GroupPivotTableDataAccess |
| GroupPivotTableActiveFieldAccess |
| GroupPivotChartFilterAndSort |
| GroupPivotChartTools |
| GroupPivotChartShowHide |
| GroupPivotChartDataAccess |
| GroupPivotChartActiveFieldAccess |
| GroupPivotChartType |
| ControlMarginsGallery |
| ControlPaddingGallery |
| AutoFormatGallery |
| PivotMoveField |
| PivotChartSortByTotalMenu |
| SharePointListsDiscardChangesMenu |
| ReviewCompareMenu |
| ReviewCompareTwoVersions |
| ReviewCombineRevisions |
| ReviewCompareMajorVersion |

| imageMso |
|------------------------------------|
| ReviewCompareLastVersion |
| ReviewCompareSpecificVersion |
| PropertyInsert |
| ObjectsAlignLeftSmart |
| ObjectsAlignRightSmart |
| ObjectsAlignTopSmart |
| ObjectsAlignBottomSmart |
| ObjectsAlignCenterHorizontalSmart |
| ObjectsAlignMiddleVerticalSmart |
| AlignDistributeHorizontally |
| AlignDistributeVertically |
| GroupProtect |
| IndexUpdate |
| MailMergeFinishAndMergeMenu |
| HyphenationMenu |
| MailMergeRules |
| SlideShowCustomMenu |
| SlideShowResolutionGallery |
| SlideShowUseRehearsedTimings |
| SlideShowShowPresentationOnGallery |
| GroupMonitors |
| MailMergeStartMailMergeMenu |
| MailMergeStartLetters |
| MailMergeStartEmail |
| MailMergeStartEnvelopes |
| MailMergeStartLabels |
| MailMergeStartDirectory |
| MailMergeClearMergeType |
| MailMergeSelectRecipients |
| TableOfContentsAddTextGallery |

| imageMso |
|---------------------------------------|
| FunctionsRecentlyUsedtInsertGallery |
| FunctionsFinancialInsertGallery |
| FunctionsDateTimeInsertGallery |
| FunctionsMathTrigInsertGallery |
| FunctionsTextInsertGallery |
| FunctionsLogicalInsertGallery |
| FunctionsStatisticalInsertGallery |
| FunctionsLookupReferenceInsertGallery |
| FunctionsInformationInsertGallery |
| GroupTableStyleOptions |
| GroupTableExternalData |
| GroupEditingExcel |
| BorderColorPicker |
| TranslationToolTip |
| FileCompatibilityChecker |
| FollowUpComposeMenu |
| ThemeFontsGallery |
| ThemeEffectsGallery |
| PrintColumns |
| PrintDataOnly |
| GroupControlAlignment |
| RmsNavigationBarHome |
| FileProperties |
| ForwardAttach |
| BackAttach |
| ManageAttachments |
| GroupPrintPreviewPrint |
| GroupPrintPreviewPreview |
| GroupPlay |
| GroupMovieOptions |

| |
|------------------------------------|
| imageMso |
| GroupSoundOptions |
| SoundPlaySoundGallery |
| SlideShowVolume |
| BreaksGallery |
| LineNumbersMenu |
| GroupOutliningClose |
| GroupOutliningTools |
| GroupMasterDocument |
| TableSelectMenu |
| TableDeleteRowsAndColumnsMenuWord |
| GroupTableMerge |
| TableAutoFitMenu |
| GroupTableDrawBorders |
| TableBordersMenu |
| SortDialog |
| SortAscendingExcel |
| SortDescendingExcel |
| SortCustomExcel |
| EditListItems |
| PublishToPdfOrEdoc |
| FileSaveAsCurrentFileFormat |
| FileSaveAsAccess2007 |
| FileSaveAsAccess2002_2003 |
| FileSaveAsAccess2000 |
| AcceptAndAdvance |
| RejectAndAdvance |
| FileCreateDocumentWorkspace |
| FileSaveToDocumentManagementServer |
| FileDocumentManagementInformation |
| ScreenNavigatorBack |

| imageMso |
|------------------------------------|
| ScreenNavigatorForward |
| FilePrepareMenu |
| FileMarkAsFinal |
| FileAddDigitalSignature |
| ForwardAsBusinessCard |
| ChooseInfoPathForm |
| RecordsTotals |
| StylesPaneNewStyle |
| ChangeCaseGallery |
| ReviewDeleteCommentsMenuPowerPoint |
| GroupMerge |
| AlignJustifyMenu |
| TextBoxInsertMenu |
| TextBoxInsertHorizontal |
| ControlProperties |
| ViewCode |
| GroupHeaderFooterLayout |
| GroupHeaderFooterInsert |
| GroupHeaderFooterNavigation |
| GroupHeaderFooterPosition |
| BlankPageInsert |
| HandoutOrientation |
| SlidesPerPageGallery |
| GroupPlaceholdersHandoutMaster |
| GroupMasterEditTheme |
| GroupMasterClose |
| GroupMasterEdit |
| GroupMasterLayout |
| GroupColorModeSetting |
| ShadowStyleGalleryClassic |

| imageMso |
|-------------------------------|
| GroupInkSelect |
| WordArtSpacingMenu |
| TextAlignMenu |
| DiagramChangeToMenuClassic |
| NumberFormatGallery |
| PictureBrightnessGallery |
| PictureContrastGallery |
| PicturePositionGallery |
| GroupPictureReset |
| GroupPictureCompress |
| GroupMailMergeStart |
| OleObjectInsertMenu |
| ViewsLayoutView |
| ForwardAsAttachment |
| InkInsertSpace |
| ShadingColorPicker |
| ShadingColorsMoreColorsDialog |
| SmartArtAddShapeAfter |
| SmartArtAddShapeBefore |
| SmartArtAddShapeAbove |
| SmartArtAddShapeBelow |
| SmartArtAddAssistant |
| ChartSwitchRowColumn |
| ChartShowData |
| ChartRefresh |
| ChartChangeType |
| GroupChartData |
| GroupChartLocation |
| GroupChartType |
| GroupTableSize |

| imageMso |
|---|
| TableCellMarginsGallery |
| SlideMasterVerticalContentPlaceholderInsert |
| _3DRotationOptionsDialog |
| _3DBevelOptionsDialog |
| SlideBackgroundStylesGallery |
| TextDirectionGallery |
| GroupTableStyleOptionsPowerPoint |
| TableStyleBandedRowsPowerPoint |
| TableStyleBandedColumnsPowerPoint |
| SelectionPane |
| SmartArtOrganizationChartLeftHanging |
| SmartArtOrganizationChartRightHanging |
| SmartArtOrganizationChartBoth |
| SmartArtOrganizationChartStandard |
| SmartArtRightToLeft |
| TableCellCustomMarginsDialog |
| ForwardContact |
| PositionAnchoringGallery |
| GroupMoveData |
| FilterAdvancedMenu |
| OmsSend |
| OmsFileSend |
| OmsSave |
| GroupOmsSend |
| GroupOmsInsert |
| GroupOmsView |
| GroupOmsMessageOptions |
| OmsViewAccountSetting |
| OmsInsertSymbolGallery |
| OmsEmoticonStringInsertGallery |

| imageMso |
|----------------------------------|
| OmsEmoticonInsertGallery |
| OmsPreviewPane |
| OmsChangeZoom |
| OmsAccountSetup |
| OmsOptions |
| OmsSlideInsert |
| OmsChangeSlideLayoutGallery |
| OmsDelete |
| OmsInsertPicture |
| OmsInsertAudio |
| OmsCustomizeLayout |
| OmsAudioFromFile |
| OmsImageFromFile |
| OmsImageFromClip |
| OmsScanImage |
| OmsNewTextMessage |
| OmsNewMultimediaMessage |
| OmsNewTextMessageToContact |
| OmsNewMultimediaMessageToContact |
| OmsForwardAsTextMessage |
| OmsForwardAsMultimediaMessage |
| OmsMaximumMessages |
| OmsGroupCreateSlides |
| ListLevelGallery |
| MultilevelListGallery |
| BuildingBlockProperties |
| ApplyStylesPane |
| NameManager |
| NameDefine |
| OpenAttachedCalendar |

| imageMso |
|-------------------------------|
| RssShareThisFeed |
| OpenInBrowserOutlook |
| InsertAlignmentTab |
| SharingOpenMailFolder |
| SharingOpenContactFolder |
| SharingOpenTaskFolder |
| SharingOpenNotesFolder |
| SharingOpenJournalFolder |
| SharingOpenDocumentFolder |
| ReplyWithInstantMessage |
| DefaultView |
| HideDetails |
| GridlinesWidthGallery |
| GridlinesColorPicker |
| GridlinesStyleGallery |
| SmartArtOrganizationChartMenu |
| SmartArtChangeLayout |
| ReadingViewToolsMenu |
| ReadingViewNextPage |
| ReadingViewPreviousPage |
| ReadingViewShowOnePage |
| ReadingViewShowTwoPages |
| ReadingViewViewOptionsMenu |
| EditWorkflowTask |
| SymbolsDialog |
| ReviewReviewingPaneHorizontal |
| ReviewReviewingPaneVertical |
| _3DEffectsGalleryClassic |
| _3DDirectionGalleryClassic |
| _3DLightingGalleryClassic |

| imageMso |
|--------------------------------------|
| GroupShadowEffects |
| Group3DEffects |
| WordArtStylesGalleryClassic |
| WordArtInsertGalleryClassic |
| TableInsertGallery |
| ShapeStylesGalleryClassic |
| WordArtChangeShapeGallery |
| ShadowColorPickerClassic |
| _3DEffectColorPickerClassic |
| ShapeFillGradientGalleryClassic |
| AsianLayoutMenu |
| JapaneseGreetingsInsertMenu |
| AlignJustifyWithMixedLanguages |
| AlignJustifyLow |
| AlignJustifyMedium |
| AlignJustifyHigh |
| AlignJustifyThai |
| ExcelSpreadsheetInsert |
| ParagraphIndentLeft |
| ParagraphIndentRight |
| ParagraphSpacingBefore |
| ParagraphSpacingAfter |
| HeaderFooterPositionHeaderFromTop |
| HeaderFooterPositionFooterFromBottom |
| ControlsGallery |
| ShapeHeight |
| ShapeWidth |
| TransitionTimeAutomaticallyAfter |
| SoundMaximumFileSize |
| CDAudioStartTrack |

| imageMso |
|----------------------------------|
| CDAudioStopTrack |
| CDAudioStartTime |
| CDAudioStopTime |
| SlideOrientationGallery |
| TextHighlightColorPicker |
| PageColorPicker |
| GoToHeader |
| GoToFooter |
| UnderlineGallery |
| UnderlineColorPicker |
| TextDirectionGalleryWord |
| GroupAddInsMenuCommands |
| GroupAddInsToolbarCommands |
| ViewFreezePanelsGallery |
| GroupInk |
| GroupInkPens |
| GroupInkClose |
| InkBallpointPen |
| InkFeltTipPen |
| InkHighlighter |
| PhoneticGuideMenu |
| PageScaleToFitScale |
| GroupBorder |
| PictureRecolorGalleryWord |
| _3DSurfaceMaterialGalleryClassic |
| _3DExtrusionDepthGalleryClassic |
| GroupHeaderFooter |
| GroupPageLayoutSetup |
| GroupPageBackground |
| FunctionsCubeInsertGallery |

| imageMso |
|-----------------------------------|
| FunctionsEngineeringInsertGallery |
| ThemeSaveCurrentPowerPoint |
| ThemeSaveCurrent |
| TextAlignGallery |
| ThemesGallery |
| CharacterSpacingGallery |
| ChartFormatDataSeries |
| ChartFormatDataLabels |
| ChartResetToMatchStyle |
| ChartSeriesTypeChange |
| Chart3DView |
| ChartFormatDataPoint |
| ChartFormatDataLabel |
| ChartFormatAxis |
| ChartFormatGridlines |
| ChartFormatDisplayUnit |
| ChartFormatLegendEntry |
| ChartFormatChartArea |
| ChartFormatLegend |
| ChartFormatErrorBars |
| ChartFormatUpBars |
| ChartFormatDownBars |
| ChartFormatHighLowLine |
| ChartFormatDropLines |
| ChartFormatTrendline |
| ChartFormatTrendlineLabel |
| ChartFormatSeriesLine |
| ChartFormatDataTable |
| ChartFormatAxisTitle |
| ChartFormatChartTitle |

| imageMso |
|-----------------------------------|
| ChartFormatFloor |
| ChartFormatSideWall |
| ChartFormatBackWall |
| ChartFormatWalls |
| ChartFormatPlotArea |
| ObjectSizeAndPositionDialog |
| ObjectSizeAndPropertiesDialog |
| ObjectSizeDialog |
| ShapeConvertToFreeform |
| AutoTextGallery |
| TextBoxInsertGallery |
| PageNumbersInHeaderInsertGallery |
| PageNumbersInFooterInsertGallery |
| PageNumbersInMarginsInsertGallery |
| SaveSelectionToAutoTextGallery |
| SaveSelectionToTextBoxGallery |
| BuildingBlocksCreateLayout |
| SaveSelectionToPageNumberTop |
| SaveSelectionToPageNumberBottom |
| SaveSelectionToPageNumberMargin |
| BuildingBlocksSaveCoverPage |
| BuildingBlocksSaveHeader |
| BuildingBlocksSaveFooter |
| BuildingBlocksSavePageNumTop |
| BuildingBlocksSavePageNumBottom |
| BuildingBlocksSavePageNum |
| HeaderFooterEditHeader |
| TableStyleBandedRowsWord |
| TableStyleBandedColumnsWord |
| MSWordApplyTableStyle |

| imageMso |
|---|
| TableStyleModify |
| TableStylesGalleryWord |
| ReviewViewChangesInTheSourceDocument |
| GroupThemesWord |
| OfficeDiagnostics |
| WindowsSwitch |
| LayoutOptionsDialog |
| DrawingObjectFormatDialog |
| FontAlternateFillBackColorPicker |
| GroupGroupingAndTotals |
| GroupFindAccess |
| GroupSchemaTools |
| GroupAdpDiagramShowHide |
| GroupAdpDiagramLayout |
| GroupAdpQueryTools |
| GroupAdpOutputOperations |
| GroupAdpQueryType |
| GroupAdpSqlStatementDesignTools |
| ConvertToSmartArt |
| ConvertToSmartArtMoreSmartArtGraphicsDialog |
| ReflectionGallery |
| PictureRecolorGallery |
| SmartArtPromote |
| SmartArtDemote |
| TableEffectsCellBevelGallery |
| GroupPivotChartShowOrHide |
| GroupPivotChartData |
| GroupPivotChartActiveField |
| PivotChartRefresh |
| PivotChartFilterShow |

| imageMso |
|-----------------------------------|
| TableBackgroundGallery |
| _3DPerspectiveIncrease |
| _3DPerspectiveDecrease |
| ChartSecondaryHorizontalAxisTitle |
| ChartSecondaryVerticalAxisTitle |
| ChartSecondaryHorizontalGridlines |
| ChartSecondaryVerticalGridlines |
| ChartSecondaryHorizontalAxis |
| ChartSecondaryVerticalAxis |
| ReadingModeToPrintView |
| GroupAddInsCustomToolbars |
| MoviePlayAutomatically |
| ReviewDeleteCommentsMenu |
| ConditionalFormattingMenu |
| GroupPersonalInfo |
| GroupMessageOptions |
| GroupAttach |
| GroupNames |
| GroupRespond |
| GroupMembers |
| GroupContactOptions |
| GroupJunkEmail |
| ForwardMenu |
| JunkEmailSafeListsMenu |
| UseVotingButtonsMenu |
| RecentlyUsedFolder |
| ArrangeWindowsDialog |
| GroupForm |
| MakeSameSizeMenu |
| CategorizeMenu |

| imageMso |
|--|
| ReplyAllWithInstantMessage |
| ObjectBringToFrontMenu |
| ObjectSendToBackMenu |
| ObjectsGroupMenu |
| SignatureLineInsertMenu |
| FileSaveAsWord97_2003 |
| ViewDisplayInColor |
| HyperlinkFlyoutAnchor |
| FileSaveAsPowerPoint97_2003 |
| SelectionPaneHidden |
| PivotFieldList |
| FilePackageAndSign |
| TotalsMenu |
| EnglishWritingAssistant |
| TableOfContentsGallery |
| SaveSelectionToTableOfContentsGallery |
| BuildingBlocksCreateTableOfFigures |
| BuildingBlocksCreateTableOfAuthorities |
| BuildingBlocksCreateIndex |
| BuildingBlocksSaveTableOfContents |
| BuildingBlocksSaveTableOfFigures |
| BuildingBlocksSaveTableOfAuthorities |
| BuildingBlocksSaveIndex |
| FileSaveAsExcel97_2003 |
| TextBoxInsertExcel |
| TextBoxStyleGallery |
| TableColumnWidth |
| TableRowHeight |
| TextBoxPositionGallery |
| FilePermissionRestrictMenu |

| |
|---------------------------------------|
| imageMso |
| GroupEnvelopeLabelCreate |
| ReviewInkCommentNew |
| InsertCellstMenu |
| PivotTableInsertMenu |
| OutlineGroupMenu |
| OutlineUngroupMenu |
| FormulaMoreFunctionsMenu |
| WordInsertTableOfContents2 |
| ReadingViewTrackChanges |
| ReadingViewShowCommentsAndChangesMenu |
| ReadingViewChangeInkPenMenu |
| DocumentPanelTemplate |
| GroupModify |
| ObjectSizeDialogClassic |
| ViewGridlinesToggleExcel |
| TextBoxInsertVerticalWord |
| BevelShapeGallery |
| _3DBevelPictureTopGallery |
| _3DMaterialMixed |
| _3DMaterialPlastic |
| _3DMaterialMetal |
| BibliographyAddNewPlaceholder |
| TableStyleClearTable |
| GroupTable |
| GroupTableCellSize |
| TableEffectsMenu |
| EditLinks |
| PositionAbsoluteMarks |
| QuerySplitMenu |
| FileSaveAsMenuAccess |

| imageMso |
|-------------------------------------|
| FileServerMenu |
| GlowColorPicker |
| RecolorColorPicker |
| GlowColorMoreColorsDialog |
| PictureRecolorMoreColorsDialog |
| ShapesDuplicate |
| CloseDocument |
| WatermarkRemove |
| CoverPageRemove |
| HeaderFooterRemoveHeaderWord |
| HeaderFooterRemoveFooterWord |
| PageNumbersRemove |
| TableOfContentsRemove |
| SmartArtAddBullet |
| TableWidth |
| TableHeight |
| SlideMasterPicturePlaceholderInsert |
| SlideMasterInsertPlaceholderMenu |
| RemoveCitation |
| EditCitation |
| TableBorderColorMoreColorsDialog |
| TableFillColorMoreColorsDialog |
| PictureChange |
| GroupWordArtStyles |
| TextFillColorPicker |
| TextOutlineColorPicker |
| TextOutlineColorMoreColorsDialog |
| TextEffectsMenu |
| TextStylesGallery |
| WordArtClear |

| imageMso |
|---|
| TextPictureFill |
| TextFillGradientGallery |
| TextFillMoreGradientsDialog |
| TextFillTextureGallery |
| TextOutlineDashesGallery |
| TextOutlineMoreLinesDialog |
| TextOutlineWeightGallery |
| TextEffectShadowGallery |
| TextEffectsMoreShadowsDialog |
| TextEffectsBevelMore3DOptionsDialog |
| TextEffects3DRotationGallery |
| TextEffects3DRotationOptionsDialog |
| TextEffectGlowGallery |
| TextGlowColorPicker |
| TextGlowColorMoreColorsDialog |
| TextReflectionGallery |
| ShapeEffectsMenu |
| NewTableStyle2 |
| NewPivotTableStyle2 |
| ShapeConnectorStyleMenu |
| PivotTableSubtotalsDoNotShow |
| PivotTableSubtotalsOnBottom |
| PivotTableSubtotalsOnTop |
| PivotTableGrandTotalsOffForRowsAndColumns |
| PivotTableGrandTotalsOnForRowsAndColumns |
| PivotTableGrandTotalsOnForRowsOnly |
| PivotTableGrandTotalsOnForColumnsOnly |
| PivotTableBlankRowsInsert |
| PivotTableBlankRowsRemove |
| ShowMessagePage |

| imageMso |
|------------------------------|
| ShowCustomPage |
| ShowTrackingPage |
| ShowCustomPropertiesPage |
| ShowCustomActionsPage |
| ShowFormRegionPage |
| GroupShow |
| ShowAllFieldsPage |
| MessageToAttendeesMenu |
| ReminderGallery |
| ShowTimeAsGallery |
| ProposeNewTimeMenu |
| GroupAttendeesMeetingNotSent |
| GroupAppointmentOptions |
| ShowAppointmentPage |
| ShowSchedulingPage |
| ShowContactPage |
| ShowCertificatesPage |
| ShowActivitesPage |
| ShowDetailsPage |
| ShowMembersPage |
| ShowNotesPage |
| GroupCopy |
| InterconnectOpen |
| InterconnectNextSide |
| DefinedPrintStyle |
| ItemProperties |
| PermissionRestrictMenu |
| MoveItem |
| ShowSharePage |
| EditBusinessCard |

| |
|-------------------------------|
| imageMso |
| GroupInterconnect |
| ContactPictureMenu |
| ContactSendMenu |
| DistributionListSelectMembers |
| DistributionListAddNewMember |
| DistributionListRemoveMember |
| DistributionListUpdateMembers |
| ObjectsGroupOutlook |
| ObjectsUngroupOutlook |
| ObjectBringToFrontOutlook |
| ObjectSendToBackOutlook |
| ObjectBringForwardOutlook |
| ObjectSendBackwardOutlook |
| FormPublishMenu |
| GroupDesign |
| PageMenu |
| AppointmentBusy |
| AppointmentOutOfOffice |
| ShowTaskPage |
| ShowTaskDetailsPage |
| GroupManageTask |
| GroupTaskOptions |
| ShowJournalPage |
| StartTimer |
| PauseTimer |
| FormRegionMenu |
| ObjectsAlignMenuOutlook |
| GroupSend |
| IMMenu |
| UpgradeDocument |

| imageMso |
|-------------------------------|
| GroupHeaderFooterClose |
| GroupChartCurrentSelection |
| GroupChartLabels |
| AlignTopExcel |
| AlignMiddleExcel |
| AlignBottomExcel |
| CustomActionsMenu |
| GroupActions |
| TableOfAuthoritiesUpdate |
| TableOfFiguresUpdate |
| PrintPreviewMultiplePagesMenu |
| ContentControlsGroup |
| FormsMenu |
| TablesMenu |
| ReportsMenu |
| BevelTextGallery |
| PictureCorrectionsDialog |
| PostcardWizard |
| BusinessFormWizard |
| GroupTextBoxStyles |
| GroupTableStylesWord |
| GroupWordArtStylesClassic |
| PageBorderAndShadingDialog |
| GroupTableStylesExcel |
| OutlineViewClose |
| ConditionalFormattingNewRule |
| SmartArtAddShapeSplitMenu |
| ContentControlRichText |
| ContentControlText |
| ContentControlPicture |

| imageMso |
|-------------------------------------|
| ContentControlComboBox |
| ContentControlDropDownList |
| ContentControlBuildingBlockGallery |
| ContentControlDate |
| AccountMenu |
| MoveToFolderFileMenu |
| RssDownloadContent |
| InsertRow |
| InsertColumn3 |
| MasterNotesPageOrientation |
| ViewRulerExcel |
| ReadingNextPageRtl |
| ReadingPrevPageRtl |
| AutoSummaryToolsMenu |
| GroupInkFormat |
| InkColorPicker |
| GroupDiagramLayoutClassic |
| GroupDiagramStylesClassic |
| GroupOrganizationChartLayoutClassic |
| GroupOrganizationChartStyleClassic |
| TextBoxDrawMenu |
| TextBoxInsertWord |
| GroupOrganizationChartSelect |
| BorderColorPickerExcel |
| TextDirectionMoreOptionsDialog |
| TextAlignMoreOptionsDialog |
| ParagraphMoreColumnsDialog |
| ObjectBringToFrontMenuOutlook |
| ObjectSendToBaseMenuOutlook |
| Call |

| imageMso |
|--|
| BibliographyGallery |
| CustomHeaderGallery |
| CustomFooterGallery |
| CustomCoverPageGallery |
| CustomPageNumberGallery |
| CustomPageNumberTopGallery |
| CustomPageNumberBottomGallery |
| CustomPageMargins |
| CustomWatermarkGallery |
| CustomEquationsGallery |
| CustomTablesGallery |
| CustomQuickPartsGallery |
| CustomAutoTextGallery |
| CustomTextBoxGallery |
| CustomTableOfContentsGallery |
| CustomBibliographyGallery |
| CustomGallery1 |
| CustomGallery2 |
| CustomGallery3 |
| CustomGallery4 |
| CustomGallery5 |
| SlideBackgroundReset |
| ShapeCloud |
| MailMergeReceipientsUseOutlookContacts |
| CDAudioPlayTrackAutomatically |
| FootnoteNext |
| ColorPickerCommentFill |
| GroupConvert |
| ReviewReviewingPaneMenu |
| GroupSizeClassic |

| imageMso |
|--|
| GroupPictureSizeClassic |
| GroupPictureToolsClassic |
| GalleryAllShapesAndCanvas |
| GroupShapesClassic |
| GroupSmartArtShapes |
| GroupShapeStylesClassic |
| GroupInsertText |
| Drawing1ColorPickerFill |
| ShapeOutlineColorPicker |
| Drawing1ColorPickerLineStyles |
| Drawing1ColorPickerLineStyles2 |
| Drawing1GalleryTextures |
| InsertBuildingBlocksHeaderGallery |
| InsertBuildingBlocksFooterGallery |
| InsertBuildingBlocksCommonPartsGallery |
| InsertBuildingBlocksEquationsGallery |
| Drawing1GalleryBrightness |
| Drawing1GalleryContrast |
| GroupDiagramArrangeClassic |
| GroupTextBoxArrange |
| GroupColorModeClose |
| GroupPageSetupNotesMaster |
| GroupPageSetupHandoutMaster |
| GroupPlaceholdersNotesMaster |
| GroupInsertSlides |
| InsertPicturePowerPointFlyoutAnchor |
| TableSelectMenuPowerPoint |
| TableDeleteRowsAndColumnsMenu |
| GroupCDAudioSetup |
| AccountingFormatMenu |

| imageMso |
|---|
| GroupCommentShapes |
| GroupCommentPosition |
| TripaneViewMode |
| PrintOptionsMenu |
| FileMenuSendHeader |
| FileMenuServerTasksHeader |
| ScreenNavigatorBackMenu |
| ScreenNavigatorForwardMenu |
| OutlineCollapseMenu |
| OutlineExpandMenu |
| ContentControlsGroupMenu |
| ContentControlsUngroup |
| EditCitationButton |
| GroupChartProperties |
| ControlsGalleryClassic |
| PivotTableEditDataSource |
| FileExcelServicesOptions |
| ReviewShowSourceDocumentsMenu |
| BuildingBlocksSaveEquation |
| HeaderFooterEditFooter |
| WordContentControlEditPlaceholderToggle |
| TableExportMenu |
| TableExportTableToVisioPivotDiagram |
| EquationToggle |
| EquationInsertNew |
| EquationProfessional |
| EquationLinearFormat |
| EquationNormalText |
| EquationSymbolsInsertGallery |
| EquationIntegralGallery |

| imageMso |
|--------------------------------|
| EquationFractionGallery |
| EquationRadicalGallery |
| EquationLargeOperatorGallery |
| EquationDelimiterGallery |
| EquationScriptGallery |
| EquationFunctionGallery |
| EquationAccentGallery |
| EquationLimitGallery |
| EquationOperatorGallery |
| EquationMatrixGallery |
| EquationOptions |
| GroupEquationTools |
| GroupEquationSymbols |
| GroupEquationStructures |
| EquationOptionsMenu |
| GroupBasicText |
| GroupInclude |
| GroupFields |
| GroupMessageFormat |
| GroupTracking |
| GroupMoreOptions |
| GroupCommunicate |
| GroupTimer |
| SaveSentItemsMenu |
| DelayDeliveryOutlook |
| DirectRepliesTo |
| SaveSentItemRecentlyUsedFolder |
| SaveSentItemOtherFolder |
| FindRelatedMenu |
| OtherActionsMenu |

| |
|--|
| imageMso |
| EncodingMenu |
| Drawing1GalleryRotateObject |
| MailMergeMergeFieldInsertMenu |
| ReviewDeleteAllMarkupInPresentation |
| PasteMenu |
| GroupPictureStyles |
| PictureStylesGallery |
| GroupInsertChartsExcel |
| ChartTypeColumnInsertGallery |
| ChartTypeLineInsertGallery |
| ChartTypePieInsertGallery |
| ChartTypeBarInsertGallery |
| ChartTypeAreaInsertGallery |
| ChartTypeXYScatterInsertGallery |
| ChartTypeOtherInsertGallery |
| ChartTypeAllInsertDialog |
| PivotChartClearMenu |
| ReviewAcceptChangeAndMoveToNext |
| ReviewRejectChangeAndMoveToNext |
| PictureEffectsShadowGallery |
| PictureEffectsGlowGallery |
| PictureEffectsSoftEdgesGallery |
| PictureReflectionGallery |
| PictureReflectionGalleryItem |
| PictureRotationGallery |
| ReadingViewShowOriginalOrFinalDocument |
| InterconnectDeleteCard |
| InkToolsClose |
| GroupChineseTranslation |
| CloseMasterView |

| imageMso |
|-----------------------------|
| SheetRowsInsert |
| SheetColumnsInsert |
| SheetRowsDelete |
| SheetColumnsDelete |
| JustifyVerticalFlyoutAnchor |
| JustifyLowVertical |
| JustifyMediumVertical |
| JustifyHighVertical |
| LineSpacingMenu |
| WordOpenParaAbove |
| WordCloseParaAbove |
| WordOpenParaBelow |
| WordCloseParaBelow |
| AdpNewTable |
| ViewsSwitchToDefaultView |
| GroupDocumentViews |
| GroupViewShowHide |
| GroupWindow |
| GroupWorkbookViews |
| ViewGridlines |
| ViewHeadings |
| GroupPresentationViews |
| GroupColorGrayscale |
| RefreshAllMenu |
| DataValidationMenu |
| GroupViews |
| GroupViewsShowHide |
| GroupViewZoom |
| GroupWindowAccess |
| FPTableAutoFormat |

| imageMso |
|---|
| FPTable |
| PageViewMenu |
| ShowTimeZones |
| ViewOnlineConnection |
| CLViewDialogHelpID |
| FileDocumentEncrypt |
| WordArtFormatDialog |
| MoreTextureOptions |
| TextFillColorMoreColorsDialog |
| ZoomFitToWindow |
| ZoomTwoPages |
| BorderTopNoToggle |
| BorderBottomNoToggle |
| BorderLeftNoToggle |
| BorderRightNoToggle |
| WindowSplitToggle |
| QuickPartsInsertFromOnline |
| JotShapeRectangle |
| JotShapeEllipse |
| JotShapeParallelogram |
| JotShapeDiamond |
| JotSendPdf |
| ProjectTaskDrivers |
| ProjectStatusReports |
| ProjectRecalcChangeHighlighting |
| ProjectAdvancedDesktopReporting |
| ProjectManageDeliverables |
| ProjectManageDependenciesOnDeliverables |
| JotLineColor |
| AddInsMenu |

| imageMso |
|---------------------------------|
| GroupPrintPreviewPageSetup |
| GroupPrintPageBackground |
| ShowRuler |
| ViewDirectionMenu |
| JustifyThaiVertical |
| FileEmailAsPdfEmailAttachment |
| FileEmailAsXpsEmailAttachment |
| FontFillBackColorPicker |
| GroupArrangeOutlook |
| GroupTemplates |
| SelectMenuAccess |
| GoToMenuAccess |
| GroupQueryResults |
| GroupAdminister |
| GroupTableDesignShowHide |
| GroupPrintPreviewData |
| GroupPrintPreviewClosePreview |
| GroupFormatting |
| GroupQueryClose |
| GroupMacroClose |
| ObjectsGroupMenuOutlook |
| RegionLayoutMenu |
| ResponsesMenu |
| GroupOpen |
| SpellingMenu |
| GroupPrintPreviewZoom |
| PictureEffectsMenu |
| PictureShapeGallery |
| PictureBorderColorPickerClassic |
| GroupInsertShapes |

| imageMso |
|------------------------------|
| PhotoAlbumInsertMenu |
| GroupChartBackground |
| GroupChartAnalysis |
| ZoomToSelection |
| FileNewBlogPost |
| GroupBlogPublish |
| BlogPublishMenu |
| BlogPublish |
| BlogPublishDraft |
| BlogManageAccounts |
| GroupBlogProperties |
| BlogCategories |
| BlogCategoryInsert |
| BlogCategoriesRefresh |
| BlogOpenExisting |
| GroupBlogBasicText |
| GroupBlogInsertText |
| DeleteSlideContextual |
| AutoSigInsertPictureFromFile |
| AutoSigWebInsertHyperlink |
| GroupFind |
| ViewPreviousItemMenu |
| ViewNextItemMenu |
| ContactLinkMenu |
| NewTableStyleWord |
| CacheListData |
| AccessRefreshAllLists |
| AccessOfflineLists |
| AccessOnlineLists |
| GridlinesExcel |

| imageMso |
|------------------------------------|
| UnmergeCells |
| MenuPublish |
| FileMenuPublishHeader |
| ChangeStylesMenu |
| GroupBlogInsertLinks |
| SharePointSiteRecycleBin |
| SubformMenu |
| LabelFontDialog |
| PrintDialogAccess |
| ForwardInForwardMenu |
| FileCompatibilityCheckerPowerPoint |
| SharingNone |
| AccessRecycleBin |
| FileCompatibilityCheckerWord |
| AccessRelinkLists |
| StopAnimation |
| ViewSideBySide |
| TableBackgroundPictureFill |
| GroupAttendeesMeetingSent |
| ImexRunImport |
| ImexRunExport |
| GroupTools |
| GroupFormattingControls |
| GroupControlAlignmentLayout |
| GroupDesignGridlines |
| GroupFormattingGridlines |
| GroupFieldsTools |
| GroupFontAccess |
| GroupDatasheetRelationships |
| GroupPositionLayout |

| imageMso |
|--------------------------------------|
| GroupControlPositionLayout |
| GroupControlSize |
| PrintPreviewZoomMenu |
| GroupMarginsAndPaddingControlLayout |
| RecordsDeleteColumn |
| ViewsAdpDiagramPrintPreview |
| UpgradePresentation |
| ChartShowDataContexttualMenu |
| ChartSourceDataContexttualMenu |
| FileSaveAsOtherFormats |
| FileSaveAsWordDocx |
| FileSaveAsWordDotx |
| FileSaveAsExcelXlsx |
| FileSaveAsExcelXlsxMacro |
| FileSaveAsExcelXlsb |
| FileSaveAsPowerPointPptx |
| FileSaveAsPowerPointPpsx |
| LockCell |
| PasteAsPictureMenu |
| GroupPivotActions |
| PivotTableSelectFlyout |
| ZoomCurrent100 |
| Drawing1ColorPickerLineStylesWordArt |
| Drawing1ColorPickerFillWordArt |
| FontDialogPowerPoint |
| PowerPointParagraphDialog |
| PowerPointPageSetup |
| GroupDrawing2 |
| ShapeQuickStylesHome |
| GalleryAllShapesAndTextboxes |

| imageMso |
|------------------------------|
| TableTextStylesGallery |
| GroupTextStylesTable |
| UpdateBibliography |
| ExportToAccess |
| BizBarPublishToSharePoint |
| BlogInsertCategories |
| NewMailMessageNumbered |
| NewAppointmentNumbered |
| NewMeetingRequestNumbered |
| NewContactNumbered |
| NewDistributionListNumbered |
| NewTaskNumbered |
| NewNoteNumbered |
| FileMenuSaveAsHeaderOutlook |
| FileMenuPrintHeaderOutlook |
| GroupRecover |
| SmartArtInsertBulletRTL |
| NewContactMenu |
| GroupTableLayout |
| ObjectsArrangeMenu |
| TextFillMoreTextures |
| FileMenuPrintHeaderAccess |
| ConvertDatabaseFormat |
| PivotShowOnlyTheTopMenu |
| PivotShowOnlyTheBottomMenu |
| GroupMacros |
| PlayMacro |
| MenuMacros |
| SharingOpenWssDocumentList |
| SharingOpenWssDiscussionList |

| imageMso |
|---------------------------------|
| SharingOpenWssCalendar |
| SharingOpenWssContactList |
| SharingOpenWssTaskList |
| SharingOpenRssFeed |
| SharingOpenPublishedCalendar |
| SharingOpenWebCalendar |
| SharingOpenICalendar |
| SharingPreviewPublishedCalendar |
| SharingPreviewWssCalendar |
| SharingPreviewWssContactList |
| SharingPreviewWssTaskList |
| SharingPreviewWssDocumentList |
| SharingPreviewWssDiscussionList |
| SaveObjectAs |
| AdvertisePublishAs |
| UpgradeWorkbook |
| ReviewProtectDocumentMenu |
| ReviewProtectWorkbookMenu |
| GroupPermission |
| ReviewProtectPresentationMenu |
| BlogHomePage |
| GroupBlogProofing |
| GroupBlogStyles |
| EquationEdit |
| AlternativeText |
| ChartFormatLeaderLines |
| ThemeBrowseForThemes |
| FilePublishToSharePoint |
| FileCheckOutDiscard |
| GroupBlogSymbols |

4 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Office 2010 suites

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

5 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

6 Index

C

[Change tracking](#) 552

Custom UI
[parts](#) 6

E

[Elements](#) 8

- [box \(box grouping container\)](#) 8
- [button \(button inside of a split button\)](#) 32
- [button \(button\)](#) 12
- [button \(unsized button\)](#) 23
- [buttonGroup \(button grouping container\)](#) 41
- [checkBox \(check box\)](#) 45
- [comboBox \(combo box\)](#) 55
- [command \(repurposed command\)](#) 67
- [commands \(list of repurposed commands\)](#) 69
- [contextualTabs \(list of contextual tab sets\)](#) 69
- [control \(control clone\)](#) 78
- [control \(Quick Access Toolbar control clone\)](#) 88
- [control \(unsized control clone\)](#) 70
- [customUI \(custom UI document root\)](#) 97
- [dialogBoxLauncher \(dialog box launcher\)](#) 98
- [documentControls \(list of document-specific Quick Access Toolbar controls\)](#) 99
- [dropDown \(drop-down control\)](#) 100
- [dynamicMenu \(dynamic menu\)](#) 123
- [dynamicMenu \(unsized dynamic menu\)](#) 113
- [editBox \(edit box\)](#) 134
- [gallery \(gallery\)](#) 144
- [gallery \(unsized gallery\)](#) 161
- [group \(group\)](#) 178
- [item \(selection item\)](#) 186
- [labelControl \(text label\)](#) 188
- [menu \(dynamic menu root XML element\)](#) 229
- [menu \(menu with title\)](#) 207
- [menu \(menu\)](#) 218
- [menu \(unsized menu\)](#) 197
- [menuSeparator \(menu separator\)](#) 231
- [officeMenu \(Office menu\)](#) 235
- [qat \(Quick Access Toolbar\)](#) 236
- [ribbon \(ribbon\)](#) 237
- [separator \(separator\)](#) 238
- [sharedControls \(list of shared Quick Access Toolbar controls\)](#) 241
- [splitButton \(split button with title\)](#) 251
- [splitButton \(split button\)](#) 259
- [splitButton \(unsized split button\)](#) 242
- [tab \(tab\)](#) 269
- [tabs \(list of tabs\)](#) 273
- [tabSet \(contextual tab set\)](#) 274
- [toggleButton \(toggle button inside of a split button\)](#) 295
- [toggleButton \(toggle button\)](#) 285
- [toggleButton \(unsized toggle button\)](#) 275

G

[Glossary](#) 5

I

[Informative references](#) 5

[Introduction](#) 5

N

[Normative references](#) 5

P

Parts

[Quick Access Toolbar customizations part](#) 6

[ribbon extensibility part](#) 7

[Product behavior](#) 551

R

References

[informative](#) 5

[normative](#) 5

S

[Simple types](#) 305

[ST_BoxStyle \(box style\)](#) 305

[ST_Delegate \(callback function name\)](#) 305

[ST_GalleryItemWidthHeight \(gallery item width or height\)](#) 308

[ST_GalleryRowColumnCount \(gallery row or column count\)](#) 308

[ST_ID \(control ID\)](#) 309

[ST_ItemSize \(menu item size\)](#) 310

[ST_Keytip \(key tip\)](#) 310

[ST_LongString \(long string\)](#) 311

[ST_QID \(qualified control ID\)](#) 312

[ST_Size \(control size\)](#) 313

[ST_String \(short string\)](#) 314

[ST_StringLength \(string length\)](#) 315

[ST_UniqueID \(custom control ID\)](#) 315

[ST Uri \(image relationship ID\)](#) 316

T

[Tracking changes](#) 552