

[MS-CUSTOMUI]:

Custom UI XML Markup Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
1/15/2009	1.0	Major	Initial Availability
7/13/2009	1.01	Major	Revised and edited the technical content
8/28/2009	1.02	Editorial	Revised and edited the technical content
11/6/2009	1.03	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.5	Minor	Clarified the meaning of the technical content.
4/11/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.5	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	2.5	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.5	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.5	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
7/31/2014	2.6	Minor	Clarified the meaning of the technical content.
10/30/2014	3.0	Major	Significantly changed the technical content.
3/16/2015	4.0	Major	Significantly changed the technical content.
9/4/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/15/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/17/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/10/2017	5.0	Major	Significantly changed the technical content.
9/19/2017	6.0	Major	Significantly changed the technical content.
4/27/2018	7.0	Major	Significantly changed the technical content.
8/28/2018	8.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	6
1.2.1	Normative References	7
1.2.2	Informative References	7
2	Custom UI	8
2.1	Parts	8
2.1.1	Quick Access Toolbar Customizations Part	8
2.1.2	Ribbon Extensibility Part	9
2.2	Elements	10
2.2.1	box (Box Grouping Container)	10
2.2.2	button (Button)	14
2.2.3	button (Unsize Button)	24
2.2.4	button (Button Inside of a Split Button)	33
2.2.5	buttonGroup (Button Grouping Container)	41
2.2.6	checkBox (Check Box)	45
2.2.7	comboBox (Combo Box)	54
2.2.8	command (Repurposed Command)	66
2.2.9	commands (List of Repurposed Commands)	67
2.2.10	contextualTabs (List of Contextual Tab Sets)	68
2.2.11	control (Unsize Control Clone)	68
2.2.12	control (Control Clone)	76
2.2.13	control (Quick Access Toolbar Control Clone)	86
2.2.14	customUI (Custom UI Document Root)	95
2.2.15	dialogBoxLauncher (Dialog Box Launcher)	95
2.2.16	documentControls (List of Document-Specific Quick Access Toolbar Controls)	96
2.2.17	dropDown (Drop-down Control)	97
2.2.18	dynamicMenu (Unsize Dynamic Menu)	109
2.2.19	dynamicMenu (Dynamic Menu)	118
2.2.20	editBox (Edit Box)	129
2.2.21	gallery (Gallery)	138
2.2.22	gallery (Unsize Gallery)	154
2.2.23	group (Group)	168
2.2.24	item (Selection Item)	176
2.2.25	labelControl (Text Label)	178
2.2.26	menu (Unsize Menu)	187
2.2.27	menu (Menu with Title)	196
2.2.28	menu (Menu)	206
2.2.29	menu (Dynamic Menu Root XML Element)	217
2.2.30	menuSeparator (Menu Separator)	219
2.2.31	officeMenu (Office Menu)	222
2.2.32	qat (Quick Access Toolbar)	223
2.2.33	ribbon (Ribbon)	224
2.2.34	separator (Separator)	225
2.2.35	sharedControls (List of Shared Quick Access Toolbar Controls)	228
2.2.36	splitButton (Unsize Split Button)	229
2.2.37	splitButton (Split Button with Title)	237
2.2.38	splitButton (Split Button)	246
2.2.39	tab (Tab)	255
2.2.40	tabs (List of Tabs)	259
2.2.41	tabSet (Contextual Tab Set)	260
2.2.42	toggleButton (Unsize Toggle Button)	261
2.2.43	toggleButton (Toggle Button)	270
2.2.44	toggleButton (Toggle Button Inside of a Split Button)	280

2.3	Simple Types.....	289
2.3.1	ST_BoxStyle (Box Style).....	289
2.3.2	ST_Delegate (Callback Function Name)	290
2.3.3	ST_GalleryItemWidthHeight (Gallery Item Width or Height)	292
2.3.4	ST_GalleryRowColumnCount (Gallery Row or Column Count)	293
2.3.5	ST_ID (Control ID)	293
2.3.6	ST_ItemSize (Menu Item Size).....	294
2.3.7	ST_Keytip (Keytip)	295
2.3.8	ST_LongString (Long String)	295
2.3.9	ST_QID (Qualified Control ID)	296
2.3.10	ST_Size (Control Size)	298
2.3.11	ST_String (Short String).....	298
2.3.12	ST_StringLength (String Length)	299
2.3.13	ST_UniqueID (Custom Control ID)	300
2.3.14	ST Uri (Image Relationship ID).....	300
3	Appendix A: Custom UI Control ID Tables	302
3.1	idMso Tables	302
3.1.1	Word 2007.....	302
3.1.2	Excel 2007.....	356
3.1.3	PowerPoint 2007	391
3.2	imageMso Table.....	417
4	Appendix B: Full XML Schemas	520
4.1	http://schemas.microsoft.com/office/2006/01/customui Schema	520
5	Appendix C: Product Behavior	553
6	Change Tracking.....	554
7	Index.....	555

1 Introduction

In creating an interoperable implementation, it is 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 to implement UI customization in the context of ECMA-376 Office Open XML File Formats, as described in [\[ECMA-376\]](#).

1.1 Glossary

This document uses the following terms:

add-in: Supplemental functionality that is provided by an external application or macro to extend the capabilities of an application.

KeyTip: A small, pop-up window that appears over commands on the ribbon when users press the ALT key. By pressing the key that is displayed in a KeyTip, users can execute the command that is associated with the KeyTip.

macro: A set of instructions that are recorded or written, and then typically saved to a file. When a macro is run, all of the instructions are performed automatically.

XML fragment: Lines of text that adhere to XML tag rules, as described in [\[XML\]](#), but do not have a Document Type Definition (DTD) or schema, processing instructions, or any other header information.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [\[XML\]](#).

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring **XML schemas**.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

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.

[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>

[MS-CUSTOMUI2] Microsoft Corporation, "[Custom UI XML Markup Version 2 Specification](#)".

[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>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

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, as specified in [\[ECMA-376\]](#). 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

The syntax of the structures contained in this part uses **XML schema definition (XSD)**, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#).

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#).

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

For example, a user can customize the quick access toolbar for his WordProcessingML document to contain the UI controls that they commonly use.

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

For 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>
```

The root element for a part of this content type is **customUI**.

For example, the following Quick Access Toolbar Customizations content markup specifies that the control with identifier "SpellingAndGrammar" is to 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>

```

A Quick Access Toolbar Customizations part is located within the package containing the source relationship. Expressed syntactically, the **TargetMode** attribute of the **Relationship** element is "Internal".

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

2.1.2 Ribbon Extensibility 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/extensibility

The syntax of the structures contained in this part uses **XML schema definition (XSD)**, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#).

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#).

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

For example, a SpreadsheetML document that represents a timecard could contain custom UI controls to guide the user in filling out the timecard.

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

For 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>

```

The root element for a part of this content type is **customUI**.

For example, the following Ribbon Extensibility content markup specifies that the ribbon tab with identifier "TabHome" is to 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>
```

A Ribbon Extensibility part is located within the package containing the source relationship. Expressed syntactically, the **TargetMode** attribute of the **Relationship** element is "Internal".

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

- Image Part, as specified in [\[ECMA-376\]](#) Part 1 section 15.2.13.

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, such as hiding or disabling built-in UI controls or repurposing command actions.
- Creation of custom UI controls, such as a custom ribbon tab, menu item, or quick access toolbar button.

For 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 identifier of "Bold", hides the ribbon tab with an identifier of "TabHome", and creates a new custom ribbon tab with a custom button in it.

The full XML Schema Definition of the XML Schema fragments listed in this section is defined in Appendix A of [\[MS-CUSTOMUI2\]](#).

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.

For example, consider a group of controls that are laid out horizontally, as follows:

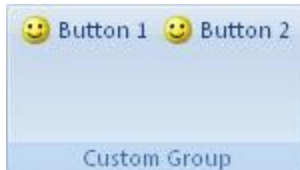


Figure 1: Controls grouped horizontally

This layout is 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 that is used if the **box** element is not specified.

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
box	2.2.1
group	2.2.23

The following table summarizes the child elements of this element.

Child Elements	Section
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
control (Control Clone)	2.2.12
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
splitButton (Split Button)	2.2.38
toggleButton (Toggle Button)	2.2.43

The following table summarizes the attributes of this element.

Attributes	Description
boxStyle (box style)	<p>Specifies the layout direction for the child controls inside of the box element. If this attribute is omitted, the child controls SHOULD be laid out horizontally. For example, consider a group of controls to be laid out vertically. This is specified using the following XML:</p> <pre data-bbox="493 352 930 422"> <box id="box" boxStyle="vertical"> ... </box> </pre> <p>The possible values for this attribute are defined by the ST_BoxStyle simple type, as specified in section 2.3.1.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function that is 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. For example, consider the following XML fragment:</p> <pre data-bbox="493 726 1146 747"> <button id="button" getVisible="IsButtonVisible" /> </pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>The id and idQ attributes are mutually exclusive. At least one of these attributes is to be specified. For example, consider the following XML fragment:</p> <pre data-bbox="493 1136 993 1157"> <button id="MyButton" label="Button" /> </pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idQ (qualified control identifier)	<p>Specifies a qualified identifier 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 and idQ attributes are mutually exclusive. At least one of these attributes is to be specified. For example, consider the following XML fragment:</p> <pre data-bbox="493 1528 1284 1814"> <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 data-bbox="496 201 638 222"></customUI></p> <p data-bbox="444 279 1442 354">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="444 363 1442 415">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 436 406 541">insertAfterMso (identifier of built-in control to insert after)</p>	<p data-bbox="444 436 1386 489">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="444 497 1425 573">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 data-bbox="444 581 992 606">For example, consider the following XML fragment:</p> <pre data-bbox="496 636 1260 705"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p data-bbox="444 758 1411 810">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an id of "TabHome".</p> <p data-bbox="444 819 1421 871">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 894 406 999">insertAfterQ (qualified identifier of control to insert after)</p>	<p data-bbox="444 894 1401 947">Specifies the qualified identifier of a control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="444 955 1425 1031">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 data-bbox="444 1039 992 1064">For example, consider the following XML fragment:</p> <pre data-bbox="496 1094 1273 1163"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p data-bbox="444 1215 1425 1268">In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="444 1276 1438 1329">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 1352 406 1457">insertBeforeMso (identifier of built-in control to insert before)</p>	<p data-bbox="444 1352 1406 1404">Specifies the identifier of a built-in control that this control is to be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="444 1413 1425 1488">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 data-bbox="444 1497 992 1522">For example, consider the following XML fragment:</p> <pre data-bbox="496 1551 1273 1621"> <tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p data-bbox="444 1673 1442 1726">In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="444 1734 1365 1787">The possible values for this attribute are defined by the ST_ID simple, as specified in section 2.3.5.</p>

Attributes	Description
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="495 405 1287 474"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</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>For example, consider the following XML fragment:</p> <pre data-bbox="495 814 993 835"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is to be hidden.</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:

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

2.2.2 button (Button)

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

For example, consider a button control, as follows:



Figure 2: A button control


This is specified using the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
box	2.2.1
group	2.2.23

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is always disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 that is called to determine the detailed description of this control. The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled)	<p>Specifies the name of a callback function that is called to determine the enabled state of this control.</p>



Attributes	Description
callback)	<p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1159 331" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is to be called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 669 1122 690" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is to be called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function that is called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1056 1146 1077" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is to be called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1413 1122 1434" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is to be called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p>



Attributes	Description
	<pre data-bbox="509 218 1224 243" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="456 296 1365 348">In this example, the GetButtonScreentip callback function is to be called when the application needs to determine the screentip of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p data-bbox="456 432 1430 485">Specifies the name of a callback function to be called to determine whether the application is to display the icon of this control.</p> <p data-bbox="456 495 1390 548">This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 558 1357 611">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 621 1000 642">For example, consider the following XML fragment:</p> <pre data-bbox="509 663 1248 688" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="456 741 1390 793">In this example, the IsButtonImageVisible callback function is to be called when the application needs to determine whether to display the icon of the button.</p> <p data-bbox="456 804 1382 856">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p data-bbox="456 873 1430 926">Specifies the name of a callback function to be called to determine whether the application is to display the label of this control.</p> <p data-bbox="456 936 1390 989">This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 999 1430 1052">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 1062 1000 1083">For example, consider the following XML fragment:</p> <pre data-bbox="509 1104 1248 1129" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="456 1182 1382 1234">In this example, the IsButtonLabelVisible callback function is to be called when the application needs to determine whether to display the label of the button.</p> <p data-bbox="456 1245 1382 1297">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSize (getSize callback)	<p data-bbox="456 1314 1422 1346">Specifies the name of a callback function to be called to determine the size of this control.</p> <p data-bbox="456 1356 1438 1409">The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p data-bbox="456 1419 1000 1440">For example, consider the following XML fragment:</p> <pre data-bbox="509 1461 1094 1486" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getSize="GetButtonSize" /></pre> <p data-bbox="456 1539 1422 1591">In this example, the GetButtonSize callback function is to be called when the application needs to determine the size of the button.</p> <p data-bbox="456 1602 1382 1654">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p data-bbox="456 1671 1382 1724">Specifies the name of a callback function to be called to determine the supertip of this control.</p> <p data-bbox="456 1734 1398 1787">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>For example, consider the following XML fragment:</p> <pre data-bbox="509 247 1198 275"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is to be called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 636 1159 663"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is to be called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control is passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1045 1008 1073"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1413 818 1440"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 220 1300 531"> <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 data-bbox="456 583 1430 684">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 695 1425 747">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 772 1442 848">Specifies the relationship identifier for an image that is to 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 data-bbox="456 856 1438 909">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 917 1005 945">For example, consider the following XML fragment:</p> <pre data-bbox="509 968 1019 993"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 1045 1425 1098">This specifies a custom button whose icon is to be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 1106 1442 1159">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 1182 1385 1266">Specifies the identifier of a built-in image that is to 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 data-bbox="456 1274 1438 1327">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1335 1005 1362">For example, consider the following XML fragment:</p> <pre data-bbox="509 1386 995 1411"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1463 1385 1491">This specifies a custom button that uses the built-in image with an identifier of "Bold".</p> <p data-bbox="456 1499 1435 1551">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1570 1401 1623">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1631 1438 1707">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 data-bbox="456 1715 1005 1743">For example, consider the following XML fragment:</p> <pre data-bbox="509 1766 1273 1816"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre>

Attributes	Description
	<p data-bbox="509 201 586 222"></tab></p> <p data-bbox="456 279 1433 327">In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 338 1433 386">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 411 407 516">insertAfterQ (qualified identifier of control to insert after)</p>	<p data-bbox="456 411 1414 464">Specifies the qualified identifier of a control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 474 1438 548">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 data-bbox="456 558 1003 579">For example, consider the following XML fragment:</p> <pre data-bbox="509 611 1284 680"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 737 1438 789">In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 800 1422 842">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 867 418 972">insertBeforeMso (identifier of built-in control to insert before)</p>	<p data-bbox="456 867 1414 919">Specifies the identifier of a built-in control that this control is to be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 930 1438 1003">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 data-bbox="456 1014 1003 1035">For example, consider the following XML fragment:</p> <pre data-bbox="509 1066 1284 1136"> <tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 1192 1433 1245">In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1255 1433 1297">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 1323 407 1428">insertBeforeQ (qualified identifier of control to insert before)</p>	<p data-bbox="456 1323 1433 1375">Specifies the qualified identifier of a control that this control is to be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1386 1438 1459">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 data-bbox="456 1470 1003 1491">For example, consider the following XML fragment:</p> <pre data-bbox="509 1522 1284 1591"> <tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 1648 1369 1701">In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 1711 1422 1753">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 1778 370 1799">keytip (keytip)</p>	<p data-bbox="456 1778 1211 1799">Specifies a string to be used as the suggested KeyTip for this control.</p>

Attributes	Description
	<p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 443 1198 468"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string that is to be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1068 768"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user. For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1300 1066"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1776 1208 1801"><button id="button" imageMso="HappyFace" label="Button"</pre>

Attributes	Description
	<pre>size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
showLabel (show label)	<p>Specifies whether this control displays its label.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified</p>

Attributes	Description
	in section 2.3.10 .
supertip (supertip)	<p>Specifies a string that is to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 814 1208 888"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1188 1105 1241"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1570 1008 1598"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an id of "TabHome" is hidden.</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:

```

<xsd:complexType name="CT_Button">
  <xsd:complexContent>
    <xsd:extension base="CT_ButtonRegular">
      <xsd:attributeGroup ref="AG_SizeAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```


2.2.3 button (Unsize Button)

This element specifies a push-button that, because of its location, cannot have its size changed. The **size** attribute is not present. This element otherwise behaves like the regular **button** element, as specified in section [2.2.2](#).

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
description	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
<p>enabled (enabled state)</p>	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 558 1300 579"><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is always disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 915 1273 936"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1299 1159 1320"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is to be called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1661 1122 1682"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is to be called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>


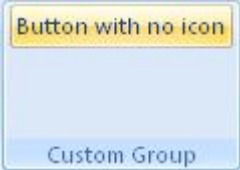
Attributes	Description
	specified in section 2.3.2.
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function that is called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1146 447"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is to be called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 783 1122 808"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is to be called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1192 1224 1218"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is to be called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application is to display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1577 1248 1602"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is to be called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel	Specifies the name of a callback function to be called to determine whether the application


Attributes	Description
(getShowLabel callback)	<p>is to display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1248 367"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is to be called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 730 1198 753"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is to be called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1115 1159 1138"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is to be called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1524 1008 1547"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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</p>

Attributes	Description
	<p>MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 285 816 310"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 680 1300 989"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier for an image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1430 1019 1455"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon is to be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier of a built-in image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 993 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="Bold" /></pre> <p data-bbox="456 296 1435 380">This specifies a custom button that uses the built-in image with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 407 1403 457">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 464 1438 541">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 data-bbox="456 548 1005 575">For example, consider the following XML fragment:</p> <pre data-bbox="509 602 1273 674" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 726 1438 777">In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 783 1435 833">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p data-bbox="456 861 1416 911">Specifies the qualified identifier of a control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 917 1438 995">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 data-bbox="456 1001 1005 1029">For example, consider the following XML fragment:</p> <pre data-bbox="509 1056 1289 1127" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1180 1419 1230">In this example, a new custom tab with an id of "MyTab" is inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 1236 1425 1287">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p data-bbox="456 1323 1419 1373">Specifies the identifier of a built-in control that this control is to be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1379 1438 1457">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 data-bbox="456 1463 1005 1491">For example, consider the following XML fragment:</p> <pre data-bbox="509 1518 1289 1589" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1642 1435 1692">In this example, a new custom tab with an id of "MyTab" is inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1698 1435 1749">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ	<p data-bbox="456 1776 1435 1803">Specifies the qualified identifier of a control that this control is to be inserted before. If the</p>

Attributes	Description
(qualified identifier of control to insert before)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1300 443"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (Keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 911 1198 934"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (Label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1213 1068 1236"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
onAction (onAction callback)	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1514 1300 1537"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>

Attributes	Description
	<p>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 627 1208 674"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1163 980 1209"><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays its label.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1457 1159 1503"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 638 1208 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 1010 1105 1058"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1394 1008 1415"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

```

</xsd:extension>
</xsd:complexContent>
</xsd: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 in the same way as the unsized **button** element, as specified in section [2.2.3](#).

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
splitButton	2.2.38
splitButton	2.2.36
splitButton	2.2.37

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is always disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
getDescription (getDescription)	<p>Specifies the name of a callback function to be called to determine the detailed description</p>



Attributes	Description
callback)	<p>of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1273 369"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 730 1159 756"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1087 1122 1113"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1472 1146 1497"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>



Attributes	Description
	<pre data-bbox="509 218 1122 241" style="text-align: center;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 296 1451 348">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 357 1385 409">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p data-bbox="456 432 1406 485">Specifies the name of a callback function that is called to determine the screentip of this control.</p> <p data-bbox="456 493 1451 569">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 data-bbox="456 577 1005 604">For example, consider the following XML fragment:</p> <pre data-bbox="509 630 1224 653" style="text-align: center;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="456 707 1427 760">In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p data-bbox="456 768 1385 821">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p data-bbox="456 842 1446 894">Specifies the name of a callback function that is called to determine whether the application SHOULD display the icon of this control.</p> <p data-bbox="456 903 1357 955">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 963 1005 991">For example, consider the following XML fragment:</p> <pre data-bbox="509 1016 1248 1039" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="456 1094 1333 1146">In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p data-bbox="456 1155 1385 1207">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p data-bbox="456 1226 1446 1278">Specifies the name of a callback function that is called to determine whether the application SHOULD display the label of this control.</p> <p data-bbox="456 1287 1430 1339">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 1348 1005 1375">For example, consider the following XML fragment:</p> <pre data-bbox="509 1400 1248 1423" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="456 1478 1443 1530">In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p data-bbox="456 1539 1385 1591">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p data-bbox="456 1610 1398 1663">Specifies the name of a callback function that is called to determine the supertip of this control.</p> <p data-bbox="456 1671 1398 1724">The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown.</p> <p data-bbox="456 1732 1005 1759">For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1198 243" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="456 296 1414 348">In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p data-bbox="456 432 1414 485">Specifies the name of a callback function that is called to determine the visibility state of this control. This attribute is prohibited.</p> <p data-bbox="456 495 1349 548">The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible.</p> <p data-bbox="456 558 1000 590">For example, consider the following XML fragment:</p> <pre data-bbox="509 600 1159 625" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 678 1446 730">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 741 1382 793">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 810 1406 894">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 905 1422 957">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 968 1000 999">For example, consider the following XML fragment:</p> <pre data-bbox="509 1010 1003 1035" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 1087 1138 1119">This specifies a custom button control with an id of "MyButton".</p> <p data-bbox="456 1129 1390 1182">The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 1199 919 1230">Specifies the identifier of a built-in control.</p> <p data-bbox="456 1241 1032 1272">The contents of this attribute are application-defined.</p> <p data-bbox="456 1283 1422 1335">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1346 1000 1377">For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 813 1413" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 1465 1114 1497">This creates a clone of the control with an identifier of "Bold".</p> <p data-bbox="456 1507 1430 1560">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 1556 919 1587">Specifies a qualified identifier for a control.</p> <p data-bbox="456 1598 1430 1650">The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p data-bbox="456 1661 1422 1713">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1724 1000 1755">For example, consider the following XML fragment:</p> <pre data-bbox="509 1766 1295 1818" style="background-color: #f0f0f0; padding: 5px;"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"</pre>

Attributes	Description
	<pre data-bbox="509 201 1105 464"> 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 data-bbox="456 520 1430 617">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 632 1425 680">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 705 1403 783">Specifies the relationship identifier for an image that is 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 data-bbox="456 791 1438 842">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 850 1005 875">For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1019 926"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 980 1438 1031">This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 1039 1438 1089">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 1117 1321 1142">Specifies the identifier of a built-in image that is used as the icon of this control.</p> <p data-bbox="456 1150 1357 1201">The contents of this attribute are application-defined and SHOULD be ignored if not understood.</p> <p data-bbox="456 1209 1438 1260">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1268 1005 1293">For example, consider the following XML fragment:</p> <pre data-bbox="509 1323 995 1344"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1398 1313 1423">This specifies a custom button that uses the built-in image with an id of "Bold".</p> <p data-bbox="456 1432 1438 1482">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1507 1403 1558">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1566 1438 1644">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 data-bbox="456 1652 1005 1677">For example, consider the following XML fragment:</p> <pre data-bbox="509 1707 1273 1749"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre>

Attributes	Description
	<p style="text-align: center;"></tab></p> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of x:OtherTab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p>

Attributes	Description
	<p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 443 1198 468"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1068 768"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user. For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1300 1066"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1776 1208 1801"><button id="button" imageMso="HappyFace" label="Button"</pre>

Attributes	Description
	<pre>size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays its label.</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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p> 

Attributes	Description
	<p>This is specified using the following XML fragment:</p> <pre data-bbox="509 254 1208 323"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 659 1105 705"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
visible (control visibility)	<p>Specifies the visibility state of the control. This attribute is prohibited.</p> <p>The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control SHOULD default to being visible.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1008 1062"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_VisibleButton">
  <xsd:complexContent>
    <xsd:restriction base="CT_ButtonRegular">
      <xsd:attribute name="visible" use="prohibited"/>
      <xsd:attribute name="getVisible" use="prohibited"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd: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.

For example, consider a group of buttons, as follows:



Figure 3: A group of buttons

This is 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>
```

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
box	2.2.1
group	2.2.23

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>

Attributes	Description
	specified in section 2.3.2 .
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>The id and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1008 474"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idQ (qualified control identifier)	<p>Specifies a qualified identifier 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 and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 842 1300 1150"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1591 1276 1661"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in</p>

Attributes	Description
	section 2.3.5 .
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an id of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</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>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 239" style="background-color: #f0f0f0; padding: 5px;"><tab idMso="TabHome" visible="false" /></pre> <p data-bbox="456 296 1406 354">In this example, the built-in tab with an identifier of "TabHome" is hidden. 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:

```

<xsd:complexType name="CT ButtonGroup">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:element name="control" type="CT_ControlCloneRegular"/>
      <xsd:element name="button" type="CT_ButtonRegular"/>
      <xsd:element name="toggleButton" type="CT_ToggleButtonRegular"/>
      <xsd:element name="gallery" type="CT_GalleryRegular"/>
      <xsd:element name="menu" type="CT_MenuRegular"/>
      <xsd:element name="dynamicMenu" type="CT_DynamicMenuRegular"/>
      <xsd:element name="splitButton" type="CT_SplitButtonRegular"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
</xsd:complexType>
```

2.2.6 checkBox (Check Box)

This element specifies a standard checkbox control.

For example, consider a checkbox control, as follows:



Figure 4: A checkbox control

This is specified using the following XML fragment:

```


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

The following table summarizes the elements that are parents of this element.

Parent Elements	Section
box	2.2.1
group	2.2.23
menu	2.2.28
menu	2.2.26
menu	2.2.29

Parent Elements	Section
menu	2.2.27
officeMenu	2.2.31

The following table summarizes the attributes of this element.


Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is always disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control. The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is</p>


Attributes	Description
	<p>specified, the control SHOULD default to being enabled. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1159 310"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 642 1122 667"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control. The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider the following XML fragment:</p> <pre data-bbox="509 1031 1146 1056"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getLabel (getLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 1122 1413"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getPressed (getPressed callback)</p>	<p>Specifies the name of a callback function to be called to determine the toggled state of this control. If this attribute is omitted, the control SHOULD default to the off state. For example, consider the following XML fragment:</p> <pre data-bbox="509 1745 1235 1770"><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre>



Attributes	Description
	<p>In this example, the IsButtonToggled callback function is called when the application needs to determine the toggle state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getScreentip (getScreentip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 533 1224 554" style="text-align: center;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 919 1247 940" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application displays the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1304 1247 1325" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1688 1198 1709" style="text-align: center;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1159 470" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getVisible="IsButtonVisible" /> </pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 856 1013 877" style="background-color: #f0f0f0; padding: 5px;"> <button id="MyButton" label="Button" /> </pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1226 818 1247" style="background-color: #f0f0f0; padding: 5px;"> <control idMso="Bold" /> </pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1617 1299 1785" style="background-color: #f0f0f0; padding: 5px;"> <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 data-bbox="509 205 716 321"> </group> </tab> </tabs> </ribbon> </customUI> </pre> <p data-bbox="456 373 1430 478">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 485 1425 537">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 562 1442 638">Specifies the relationship identifier for an image to 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 data-bbox="456 644 1438 697">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 703 1005 730">For example, consider the following XML fragment:</p> <pre data-bbox="509 758 1019 785"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 835 1425 888">This specifies a custom button whose icon is to be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 894 1442 947">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 972 1308 999">Specifies the identifier of a built-in image to be used as the icon of this control.</p> <p data-bbox="456 1005 1357 1058">The contents of this attribute are application-defined and SHOULD be ignored if not understood.</p> <p data-bbox="456 1064 1438 1117">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1123 1005 1150">For example, consider the following XML fragment:</p> <pre data-bbox="509 1178 997 1205"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1255 1287 1283">This specifies a custom button to use the built-in image with an id of "Bold".</p> <p data-bbox="456 1289 1435 1341">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1360 1401 1413">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1419 1438 1499">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 data-bbox="456 1505 1005 1533">For example, consider the following XML fragment:</p> <pre data-bbox="509 1560 1276 1633"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 1684 1409 1736">In this example, a new custom tab with an id of "MyTab" is be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1743 1435 1795">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>

Attributes	Description
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 407 1289 478"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1289 934"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1318 1289 1390"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p> 

Attributes	Description
	<p>This is specified using the following XML fragment:</p> <pre data-bbox="509 247 1198 275"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 548 1068 575"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 848 1300 875"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1583 1208 1631"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>

Attributes	Description
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 562 980 611"><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays its label.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 898 1159 947"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1675 1208 1745"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1105 527"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1008 890"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an id of "TabHome" is hidden.</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:

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

For example, consider a combo box control, as follows:

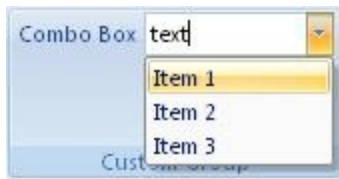


Figure 5: A combo box control

This is 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>
```

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
item (Selection Item)	2.2.24

The following table summarizes the attributes of this element.

Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1122 447"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemCount (getItemCount callback)</p>	<p>Specifies the name of a callback function to 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 that are specified as child elements. If no such items are specified, the control SHOULD be empty.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 806 1260 831"><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>In this example, the GetGalleryItemCount callback function is called when the application needs to determine the number of items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemID (getItemID callback)</p>	<p>Specifies the name of a callback function to be called to determine the identifier of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items SHOULD have empty identifiers.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1190 1222 1241"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>In this example, the GetGalleryItemID callback function is called when the application needs to determine the identifier of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemImage (getItemImage callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1575 1222 1625"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>In this example, the GetGalleryItemImage callback function is called when the application needs to determine the icon of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>


Attributes	Description
getItemLabel (getItemLabel callback)	<p>Specifies the name of a callback function to 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1224 401"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>In this example, the GetGalleryItemLabel callback function is called when the application needs to determine the label of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemScreentip (getItemScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 762 1224 810"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>In this example, the GetGalleryItemScreentip callback function is called when the application needs to determine the screentip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemSupertip (getItemSupertip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1171 1224 1220"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>In this example, the GetGalleryItemSupertip callback function is called when the application needs to determine the supertip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1577 1146 1604"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p>



Attributes	Description
callback)	<p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1122 338"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 722 1224 749"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1104 1248 1131"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1488 1248 1516"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 249 1198 275"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getText (getText callback)</p>	<p>Specifies the name of a callback function to be called to determine the text that is displayed in the control.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 575 1133 600"><editBox id="editBox" getText="GetEditBoxText" /></pre> <p>In this example, the GetEditBoxText callback function is called when the application needs to determine the text to display in the control.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 959 1159 984"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1369 1008 1394"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1736 816 1761"><control idMso="Bold" /></pre>


Attributes	Description
	<p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 516 1299 825"> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier for an image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1266 1019 1287"> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier of a built-in image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1682 995 1703"> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that uses the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in</p>

Attributes	Description
	section 2.3.5.
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1273 516"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1289 972"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1360 1289 1428"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1300 289"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="457 344 1370 394">In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="457 403 1425 453">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
invalidateContentOnDrop (invalidate content on drop)	<p data-bbox="457 480 1398 531">Specifies whether this control invalidates its contents and re-queries for them when the user opens its drop-down menu.</p> <p data-bbox="457 539 1133 564">If this attribute is omitted, its value SHOULD default to "false".</p> <p data-bbox="457 573 1005 598">For example, consider the following XML fragment:</p> <pre data-bbox="509 625 1260 697"><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p data-bbox="457 751 1425 827">In this example, this combo box clears out its items and re-calls the GetComboBoxItemCount and GetComboBoxItemLabel callback functions to populate its contents each time the user opens it.</p> <p data-bbox="457 835 1406 861">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
keytip (keytip)	<p data-bbox="457 890 1211 915">Specifies a string to be used as the suggested KeyTip for this control.</p> <p data-bbox="457 924 1347 974">The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="457 982 1086 1008">For example, consider a button with KeyTip 'K', as follows:</p>  <p data-bbox="457 1113 1002 1138">This is specified using the following XML fragment:</p> <pre data-bbox="509 1165 1198 1190"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p data-bbox="457 1245 1354 1295">The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p data-bbox="457 1320 1073 1346">Specifies a string to be used as the label for this control.</p> <p data-bbox="457 1354 1425 1404">The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="457 1413 1005 1438">For example, consider the following XML fragment:</p> <pre data-bbox="509 1465 1070 1491"><button id="button" label="Custom Button" /></pre> <p data-bbox="457 1545 1153 1570">This specifies a custom button with a label of "Custom Button".</p> <p data-bbox="457 1579 1349 1629">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
maxLength (maximum input string length)	<p data-bbox="457 1650 1406 1701">Specifies an integer to be used as the maximum length of a string that can be entered into the control.</p> <p data-bbox="457 1709 1393 1759">If the maxLength attribute is omitted, the length of the input string SHOULD NOT be limited, except by application-specific constraints.</p> <p data-bbox="457 1768 1005 1793">For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 241"><editBox id="editBox" maxLength="10" /></pre> <p data-bbox="456 296 1369 348">This specifies an edit box control that can only accept strings up to 10 characters in length.</p> <p data-bbox="456 357 1433 409">The possible values for this attribute are defined by the ST_StringLength simple type, as specified in section 2.3.12.</p>
<p data-bbox="201 432 334 510">onChange (onChange callback)</p>	<p data-bbox="456 432 1438 485">Specifies the name of a callback function to be called when the text in the control has been changed by the user.</p> <p data-bbox="456 493 1008 516">For example, consider the following XML fragment:</p> <pre data-bbox="509 543 1198 567"><editBox id="editBox" onChange="EditBoxTextChanged" /></pre> <p data-bbox="456 621 1398 674">This specifies an edit box control that calls the EditBoxTextChanged callback function when the user inputs a text string.</p> <p data-bbox="456 682 1385 735">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 756 321 808">screentip (screentip)</p>	<p data-bbox="456 756 1138 779">Specifies a string to be shown as the screentip for this control.</p> <p data-bbox="456 787 1446 867">The screentip and getScreentip 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="456 875 1099 898">For example, consider a button with a screentip, as follows:</p>  <p data-bbox="456 1251 1003 1274">This is specified using the following XML fragment:</p> <pre data-bbox="509 1302 1209 1354"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 1404 1352 1457">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1480 418 1533">showImage (show image)</p>	<p data-bbox="456 1480 959 1503">Specifies whether this control displays an icon.</p> <p data-bbox="456 1512 1357 1564">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1572 1239 1596">For example, consider a button that does not display an icon, as follows:</p>  <p data-bbox="456 1787 1003 1810">This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 980 264"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="456 317 1406 344">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
showItemImage (show item image)	<p data-bbox="456 371 1247 428">Specifies whether this control displays icons on its selection items. If this attribute is omitted, the items' icons SHOULD be shown by default.</p> <p data-bbox="456 438 1005 466">For example, consider the following XML fragment:</p> <pre data-bbox="509 491 1287 632"><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="456 686 1406 743">This specifies a gallery control that does not show any icons on its selection items. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
showLabel (show label)	<p data-bbox="456 770 1430 856">Specifies whether this control displays its label. The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 867 1005 894">For example, consider the following XML fragment:</p> <pre data-bbox="509 919 1159 966"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="456 1020 1406 1077">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p data-bbox="456 1087 1406 1115">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
sizeString (size string)	<p data-bbox="456 1127 1422 1184">Specifies a string whose size is used to determine the width of the text input area of this control.</p> <p data-bbox="456 1194 1414 1251">If this attribute is omitted, the application SHOULD determine the width of the text input area of the control automatically.</p> <p data-bbox="456 1262 1005 1289">For example, consider the following XML fragment:</p> <pre data-bbox="509 1314 1159 1341"><editBox id="editBox" sizeString="XXXXXXXXXXXX" /></pre> <p data-bbox="456 1396 1365 1453">This specifies an edit box control that SHOULD be wide enough to display the string "XXXXXXXXXXXX".</p> <p data-bbox="456 1463 1349 1491">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
supertip (supertip)	<p data-bbox="456 1512 1114 1539">Specifies a string to be shown as the supertip of the control.</p> <p data-bbox="456 1549 1398 1606">The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown.</p> <p data-bbox="456 1617 1092 1644">For example, consider a control with a supertip, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 638 1208 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 1045 1110 1096"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1432 1013 1453"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_ComboBox">
  <xsd:complexContent>
    <xsd:extension base="CT_EditBox">
      <xsd:sequence>
        <xsd:element name="item" type="CT_Item" minOccurs="0" maxOccurs="1000"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

</xsd:sequence>
<xsd:attributeGroup ref="AG_DropDownAttributes"/>
<xsd:attributeGroup ref="AG_DynamicContentAttributes"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

2.2.8 command (Repurposed Command)

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

The **enabled** and **getEnabled** attributes can be specified to disable a command.

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

For example, consider the following XML fragment:

```

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

```

In this example, the **Bold** command is permanently disabled and that the callback function **MyPasteFunction** is called when the **Paste** command is invoked.

The following table summarizes the elements that are parents of this element.

Parent Elements
commands (section 2.2.9)

The following table summarizes the attributes of this element.

Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
idMso (built-in control identifier)	<p>Specifies the identifier of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>For example, consider the following XML fragment:</p> <pre><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
onAction (onAction callback)	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>

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

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

The following table summarizes the elements that are parents of this element.

Parent Elements
customUI (section 2.2.14)

The following table summarizes the child elements of this element.

Child Elements	Subclause
command (Repurposed Command)	section 2.2.8

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

```

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

The following table summarizes the elements that are parents of this element.

Parent Elements
ribbon (section 2.2.33)

The following table summarizes the child elements of this element.

Child Elements	Subclause
tabSet (Contextual Tab Set)	section 2.2.41

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

```

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

```

2.2.11 control (Unsize Control Clone)

This element specifies a clone of a control that, because of its location, cannot have its size changed. The **size** attribute is not present. The element otherwise behaves like the regular **control** element, as specified in section [2.2.12](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the attributes of this element.


Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p>


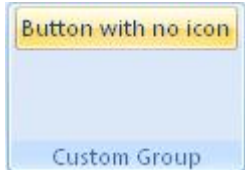
Attributes	Description
	<pre data-bbox="509 218 1300 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Disabled Button" enabled="false" /></pre> <p data-bbox="456 296 1406 380">This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 407 350 485">getEnabled (getEnabled callback)</p>	<p data-bbox="456 407 1443 457">Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p data-bbox="456 464 1386 514">The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p data-bbox="456 520 1008 550">For example, consider the following XML fragment:</p> <pre data-bbox="509 575 1159 596" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="456 653 1393 703">In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p data-bbox="456 709 1386 760">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 791 334 869">getImage (getImage callback)</p>	<p data-bbox="456 791 1430 821">Specifies the name of a callback function to be called to determine the icon of this control.</p> <p data-bbox="456 827 1438 877">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 884 1008 913">For example, consider the following XML fragment:</p> <pre data-bbox="509 938 1122 959" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="456 1016 1393 1066">In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p data-bbox="456 1073 1386 1123">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1148 334 1226">getKeytip (getKeytip callback)</p>	<p data-bbox="456 1148 1451 1199">Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p data-bbox="456 1205 1349 1255">The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 1262 1008 1291">For example, consider the following XML fragment:</p> <pre data-bbox="509 1316 1146 1337" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 1394 1393 1444">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 1451 1386 1501">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1533 431 1583">getLabel (getLabel callback)</p>	<p data-bbox="456 1533 1435 1583">Specifies the name of a callback function to be called to determine the label of this control.</p> <p data-bbox="456 1589 1429 1640">The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 1646 1008 1675">For example, consider the following XML fragment:</p> <pre data-bbox="509 1701 1122 1722" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 1778 1451 1829">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p>


Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getScreentip (getScreentip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 474 1224 495" style="text-align: center;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 856 1247 877" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1243 1247 1264" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1629 1198 1650" style="text-align: center;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>

Attributes	Description
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1159 401"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 789 1008 810"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1157 818 1178"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1549 1300 1810"><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> </customUI></pre>

Attributes	Description
	<pre data-bbox="509 201 651 247"></ribbon> </customUI></pre> <p data-bbox="456 302 1430 401">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 411 1425 464">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 489 1442 562">Specifies the relationship identifier for an image to 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 data-bbox="456 573 1438 625">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 636 1003 657">For example, consider the following XML fragment:</p> <pre data-bbox="509 684 1019 705"><button id="button" image="ForestPic" /></pre> <p data-bbox="456 762 1404 814">This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 825 1442 877">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 894 1308 915">Specifies the identifier of a built-in image to be used as the icon of this control.</p> <p data-bbox="456 926 1357 978">The contents of this attribute are application-defined and SHOULD be ignored if not understood.</p> <p data-bbox="456 989 1438 1041">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1052 1003 1073">For example, consider the following XML fragment:</p> <pre data-bbox="509 1100 993 1121"><button id="button" imageMso="Bold" /></pre> <p data-bbox="456 1178 1385 1199">This specifies a custom button that uses the built-in image with an identifier of "Bold".</p> <p data-bbox="456 1209 1433 1262">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1287 1401 1339">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1350 1438 1423">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 data-bbox="456 1434 1003 1455">For example, consider the following XML fragment:</p> <pre data-bbox="509 1482 1273 1556"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1612 1438 1665">In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1675 1433 1728">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert	<p data-bbox="456 1745 1417 1797">Specifies the qualified identifier of a control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p>

Attributes	Description
after)	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 338 1289 407"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 795 1289 865"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1255 1300 1325"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1198 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p data-bbox="456 296 1354 348">The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p data-bbox="456 373 1073 394">Specifies a string to be used as the label for this control.</p> <p data-bbox="456 405 1427 457">The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 468 1005 489">For example, consider the following XML fragment:</p> <pre data-bbox="509 516 1068 537" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Custom Button" /></pre> <p data-bbox="456 594 1153 615">This specifies a custom button with a label of "Custom Button".</p> <p data-bbox="456 625 1349 678">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
screentip (screentip)	<p data-bbox="456 703 1138 724">Specifies a string to be shown as the screentip for this control.</p> <p data-bbox="456 735 1446 814">The screentip and getScreentip 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="456 825 1099 846">For example, consider a button with a screentip, as follows:</p> <div data-bbox="456 856 951 1192" style="border: 1px solid #ccc; padding: 10px;">  </div> <p data-bbox="456 1203 1002 1224">This is specified using the following XML fragment:</p> <pre data-bbox="509 1251 1208 1297" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 1388 1349 1440">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p data-bbox="456 1461 959 1482">Specifies whether this control displays an icon.</p> <p data-bbox="456 1493 1357 1545">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1556 1239 1577">For example, consider a button that does not display an icon, as follows:</p> <div data-bbox="456 1587 699 1755" style="border: 1px solid #ccc; padding: 10px;">  </div> <p data-bbox="456 1766 1002 1787">This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 980 264"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="456 352 1406 380">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 405 407 457">showLabel (show label)</p>	<p data-bbox="456 405 1430 527">Specifies whether this control displays its label. The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 548 1159 594"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="456 653 1406 705">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p data-bbox="456 709 1406 737">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 762 418 789">supertip (supertip)</p>	<p data-bbox="456 762 1398 846">Specifies a string to be shown as the supertip of the control. The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown. For example, consider a control with a supertip, as follows:</p>  <p data-bbox="456 1272 1003 1299">This is specified using the following XML fragment:</p> <pre data-bbox="509 1325 1208 1398"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p data-bbox="456 1482 1349 1535">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1560 305 1587">tag (tag)</p>	<p data-bbox="456 1560 1390 1644">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 data-bbox="456 1648 1398 1701">If this attribute is omitted, the control's tag value SHOULD default to an empty string. For example, consider the following XML fragment:</p> <pre data-bbox="509 1728 1105 1755"><button id="button" label="Button" tag="123456"</pre>

Attributes	Description
	<pre>onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_ControlCloneRegular">
  <xsd:complexContent>
    <xsd:restriction base="CT_Control">
      <xsd:attribute name="id" use="prohibited"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd: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 cannot be cloned. 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, are copied to the clone. Location-specific properties, such as the size and visibility of the control, are not copied. These properties can be set by specifying additional attributes on the **control** element.

For example, consider the following XML fragment:

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

This results in a large copy of the **Paste** control, as follows:

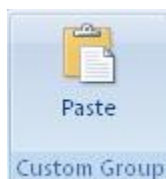



Figure 6: A Paste control

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled. This attribute cannot be used to enable a built-in control that would otherwise be disabled by the application. For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used. 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 to be called to determine the detailed description of this control. The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled. For example, consider the following XML fragment:</p>



Attributes	Description
	<pre data-bbox="509 218 1159 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="456 296 1390 348">In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 432 334 506">getImage (getImage callback)</p>	<p data-bbox="456 432 1430 516">Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 527 1003 548">For example, consider the following XML fragment:</p> <pre data-bbox="509 579 1122 600" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="456 653 1390 705">In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p data-bbox="456 716 1382 768">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 789 334 863">getKeytip (getKeytip callback)</p>	<p data-bbox="456 789 1446 842">Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p data-bbox="456 852 1344 905">The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 915 1003 936">For example, consider the following XML fragment:</p> <pre data-bbox="509 968 1146 989" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 1041 1390 1094">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 1104 1382 1157">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1167 431 1220">getLabel (getLabel callback)</p>	<p data-bbox="456 1167 1430 1220">Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 1230 1003 1251">For example, consider the following XML fragment:</p> <pre data-bbox="509 1283 1122 1304" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 1356 1446 1409">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 1419 1382 1472">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1524 367 1598">getScreentip (getScreentip callback)</p>	<p data-bbox="456 1524 1390 1577">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 1587 1446 1661">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 data-bbox="456 1671 1003 1692">For example, consider the following XML fragment:</p> <pre data-bbox="509 1724 1224 1745" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getScreentip="GetButtonScreentip" /></pre>



Attributes	Description
	<p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 562 1252 590" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application displays the label of this control.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1003 1252 1031" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSize (getSize callback)</p>	<p>Specifies the name of a callback function to be called to determine the size of this control.</p> <p>The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1360 1097 1388" style="text-align: center;"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p>


Attributes	Description
	<pre data-bbox="509 218 1198 241" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="456 296 1414 348">In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p data-bbox="456 432 1446 485">Specifies the name of a callback function to be called to determine the visibility state of this control.</p> <p data-bbox="456 495 1349 548">The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible.</p> <p data-bbox="456 558 1000 579">For example, consider the following XML fragment:</p> <pre data-bbox="509 602 1159 625" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 680 1446 732">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 743 1382 795">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 810 1406 894">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 905 1422 957">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 968 1000 989">For example, consider the following XML fragment:</p> <pre data-bbox="509 1012 1005 1035" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 1089 1216 1110">This specifies a custom button control with an identifier of "MyButton".</p> <p data-bbox="456 1121 1390 1173">The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 1199 919 1220">Specifies the identifier of a built-in control.</p> <p data-bbox="456 1230 1032 1251">The contents of this attribute are application-defined.</p> <p data-bbox="456 1262 1422 1314">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1325 1000 1346">For example, consider the following XML fragment:</p> <pre data-bbox="509 1369 818 1392" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 1457 1118 1478">This creates a clone of the control with an identifier of "Bold".</p> <p data-bbox="456 1488 1430 1541">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 1556 919 1577">Specifies a qualified identifier for a control.</p> <p data-bbox="456 1587 1430 1640">The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p data-bbox="456 1650 1422 1703">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1713 1000 1734">For example, consider the following XML fragment:</p> <pre data-bbox="509 1757 1300 1810" style="background-color: #f0f0f0; padding: 5px;"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"</pre>

Attributes	Description
	<pre data-bbox="509 201 1105 464"> 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 data-bbox="456 520 1430 617">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 632 1425 680">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 705 1442 779">Specifies the relationship identifier for an image to 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 data-bbox="456 793 1438 842">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 856 1003 877">For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1019 926"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 982 1406 1031">This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 1045 1442 1094">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 1117 1308 1138">Specifies the identifier of a built-in image to be used as the icon of this control.</p> <p data-bbox="456 1152 1357 1201">The contents of this attribute are application-defined and SHOULD be ignored if not understood.</p> <p data-bbox="456 1215 1438 1264">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1278 1003 1299">For example, consider the following XML fragment:</p> <pre data-bbox="509 1327 997 1348"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1404 1386 1425">This specifies a custom button that uses the built-in image with an identifier of "Bold".</p> <p data-bbox="456 1440 1438 1488">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1507 1403 1556">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1570 1438 1644">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 data-bbox="456 1659 1003 1680">For example, consider the following XML fragment:</p> <pre data-bbox="509 1707 1276 1749"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre>

Attributes	Description
	<p style="text-align: center;"></tab></p> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p>

Attributes	Description
	<p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 443 1198 468"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1068 768"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user. For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1300 1066"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1776 1208 1801"><button id="button" imageMso="HappyFace" label="Button"</pre>

Attributes	Description
	<pre>size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
showLabel (show label)	<p>Specifies whether this control displays its label.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
<p>supertip (supertip)</p>	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 877 1208 947"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1283 1105 1331"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1667 1008 1688"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_ControlClone">
  <xsd:complexContent>
    <xsd:restriction base="CT_Button">
      <xsd:attribute name="id" use="prohibited"/>
      <xsd:attribute name="onAction" use="prohibited"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd: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** element, as specified in section [2.2.12](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
documentControls (section 2.2.16); sharedControls (section 2.2.35)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled. This attribute cannot be used to enable a built-in control that would otherwise be disabled by the application. For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1273 447"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 806 1159 831"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1163 1122 1188"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1547 1146 1572"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified,</p>

Attributes	Description
	<p>no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1122 310"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getScreentip (getScreentip callback)</p>	<p>Specifies the name of a callback function to be called to determine the screentip of this control. 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 695 1224 720"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control. This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large". The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon. For example, consider the following XML fragment:</p> <pre data-bbox="509 1140 1252 1165"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control. This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large". The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 1581 1252 1606"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>


Attributes	Description
getSize (getSize callback)	<p>Specifies the name of a callback function to be called to determine the size of this control. The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1094 380"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 737 1198 764"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1121 1159 1148"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (custom control identifier)	<p>Specifies the identifier for a custom control. All new custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>The id, idQ, and idMso attributes are mutually exclusive.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1507 1008 1535"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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 MUST be specified.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 818 275" style="margin-left: 40px;"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier for a control. The identifier is qualified with an XML namespace prefix that specifies the owner of the control. If the namespace is equal to the Custom UI namespace, the idQ attribute behaves in the same manner as the idMso attribute. If the namespace is equal to the name of the current file, the idQ attribute behaves like the id attribute. If the namespace is equal to the name of a different file, the attribute references a control from that file.</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.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 747 1008 863" style="margin-left: 40px;"><tab idQ="x:OtherTab"> <group id="MyGroup" label="My Group"> ... </group> </tab></pre> <p>In this case x is an XML namespace equal to the name of another file that has a Custom UI document with a tab with an identifier of "OtherTab". This example adds a custom group to that tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier for an image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1278 1019 1299" style="margin-left: 40px;"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier of a built-in image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1698 997 1719" style="margin-left: 40px;"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that uses the built-in image with an identifier of "Bold".</p>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 474 1273 541"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 932 1289 999"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 1289 1455"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 249 1300 323"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 789 1198 816"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1089 1070 1117"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1211 264"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 352 1352 405">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 430 422 483">showImage (show image)</p>	<p data-bbox="456 430 1390 514">Specifies whether this control displays an icon. This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 522 1357 575">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 583 1239 609">For example, consider a button that does not display an icon, as follows:</p> <div data-bbox="461 615 699 785" data-label="Image"> </div> <p data-bbox="456 795 1003 821">This is specified using the following XML fragment:</p> <pre data-bbox="509 848 980 894"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="456 982 1406 1008">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 1031 406 1083">showLabel (show label)</p>	<p data-bbox="456 1031 1390 1115">Specifies whether this control displays its label. This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 1123 1430 1176">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 1184 1006 1209">For example, consider the following XML fragment:</p> <pre data-bbox="509 1236 1159 1283"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="456 1339 1406 1392">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p data-bbox="456 1400 1406 1425">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 1446 399 1472">size (control size)</p>	<p data-bbox="456 1446 802 1472">Specifies the size of the control.</p> <p data-bbox="456 1480 1443 1533">The size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p data-bbox="456 1541 984 1566">For example, consider a large button, as follows:</p> <div data-bbox="461 1572 618 1743" data-label="Image"> </div> <p data-bbox="456 1753 1003 1778">This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1045 243"><toggleButton idMso="Bold" size="large" /></pre> <p data-bbox="456 331 1427 384">The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
<p data-bbox="201 407 422 432">supertip (supertip)</p>	<p data-bbox="456 407 1114 432">Specifies a string to be shown as the supertip of the control.</p> <p data-bbox="456 441 1398 493">The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown.</p> <p data-bbox="456 501 1091 527">For example, consider a control with a supertip, as follows:</p>  <p data-bbox="456 919 1002 945">This is specified using the following XML fragment:</p> <pre data-bbox="509 972 1208 1045"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p data-bbox="456 1098 1349 1150">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1169 375 1222">visible (control visibility)</p>	<p data-bbox="456 1169 911 1194">Specifies the visibility state of the control.</p> <p data-bbox="456 1203 1365 1255">The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control SHOULD default to being visible.</p> <p data-bbox="456 1264 1005 1289">For example, consider the following XML fragment:</p> <pre data-bbox="509 1316 1008 1341"><tab idMso="TabHome" visible="false" /></pre> <p data-bbox="456 1394 1256 1419">In this example, the built-in tab with an identifier of "TabHome" is hidden.</p> <p data-bbox="456 1428 1406 1453">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:

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

</xsd:complexType>

2.2.14 customUI (Custom UI Document Root)

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

The following table summarizes the child elements of this element.

Child Elements	Section
commands (List of Repurposed Commands)	2.2.9
ribbon (Ribbon)	2.2.33

The following table summarizes the attributes of this element.

Attributes	Description
loadImage (loadImage callback)	<p>Specifies the name of a callback function to be called when the application needs to load an image for a control's icon.</p> <p>For example, consider the following XML fragment:</p> <pre><customUI xmlns="..." loadImage="LoadImageFunction" /></pre> <p>In this example, the LoadImageFunction callback is called to load icon images.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
onLoad (onLoad callback)	<p>Specifies the name of a callback function to be called when the Custom UI file is loaded by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><customUI xmlns="..." onLoad="OnCustomUILoaded" /></pre> <p>In this example, the OnCustomUILoaded callback function is called when the containing Custom UI file is loaded.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>

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

```
<xsd:complexType name="CT_CustomUI">
  <xsd:sequence>
    <xsd:element name="commands" type="CT Commands" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ribbon" type="CT_Ribbon" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="onLoad" type="ST Delegate" use="optional"/>
  <xsd:attribute name="loadImage" type="ST_Delegate" use="optional"/>
</xsd:complexType>
```

2.2.15 dialogBoxLauncher (Dialog Box Launcher)

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

For example, consider a dialog box launcher control, as follows:

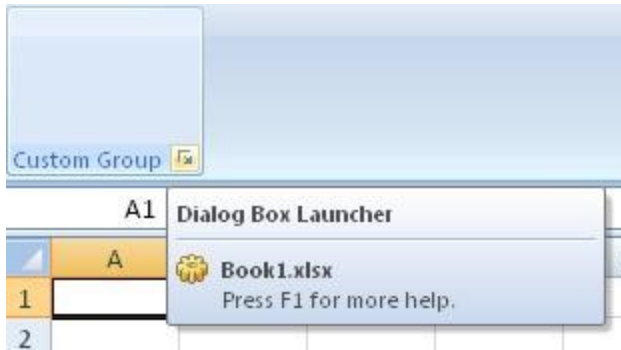


Figure 7: A dialog box launcher control

This is specified using the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
button (Unsize Button)	2.2.3

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

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

For 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 is specified using the following XML fragment:

```

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

```

The following table summarizes the elements that are parents of this element.

Parent Elements
qat (section 2.2.32)

The following table summarizes the child elements of this element.

Child Elements	Section
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:

```

<xsd:complexType name="CT_QatItems">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:element name="control" type="CT_ControlCloneQat"/>
      <xsd:element name="button" type="CT_ButtonRegular"/>
      <xsd:element name="separator" type="CT_Separator"/>
    </xsd:choice>
  </xsd:sequence>
</xsd: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 after its selection items.

For example, consider a drop-down control, as follows:

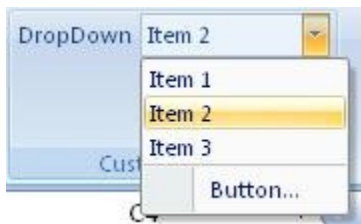


Figure 9: A drop-down control

This is 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>

```

</dropDown>

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
button (Unsize Button)	2.2.3
item (Selection Item)	2.2.24

The following table summarizes the attributes of this element.

Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemCount (getItemCount callback)</p>	<p>Specifies the name of a callback function to 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 that are specified as child elements. If no such items are specified, the control SHOULD be empty.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1260 474"><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>In this example, the GetGalleryItemCount callback function is called when the application needs to determine the number of items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemID (getItemID callback)</p>	<p>Specifies the name of a callback function to be called to determine the identifier of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items SHOULD have empty identifiers.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 831 1221 882"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>In this example, the GetGalleryItemID callback function is called when the application needs to determine the identifier of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemImage (getItemImage callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1213 1221 1264"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>In this example, the GetGalleryItemImage callback function is called when the application needs to determine the icon of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemLabel (getItemLabel callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1596 1221 1646"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>In this example, the GetGalleryItemLabel callback function is called when the application needs to determine the label of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>


Attributes	Description
getItemScreentip (getItemScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1224 428"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>In this example, the GetGalleryItemScreentip callback function is called when the application needs to determine the screentip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemSupertip (getItemSupertip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 785 1224 833"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>In this example, the GetGalleryItemSupertip callback function is called when the application needs to determine the supertip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1194 1148 1220"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1551 1122 1577"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip	<p>Specifies the name of a callback function to be called to determine the screentip of this control.</p>


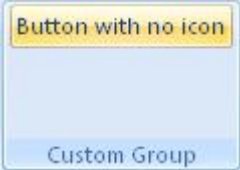
Attributes	Description
callback)	<p>The getScreentip and screentip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD display the label of the control as the screentip or display no screentip at all.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 338 1224 363"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSelectedItemID (getSelectedItemID callback)	<p>Specifies the name of a callback function to be called to determine the identifier of the item to be selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 722 1224 793"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p>In this example, the GetGallerySelectedItemID callback function is called when the application needs to determine the selected item in the gallery. In this example the callback function returns one of the identifiers returned by the GetItemID callback function.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSelectedItemIndex (getSelectedItemIndex callback)	<p>Specifies the name of a callback function to be called to determine the index of the item to be selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1184 1224 1230"><gallery id="gallery" getItemCount="GetGalleryItemCount" getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p>In this example, the GetGallerySelectedItemIndex callback function is called when the application needs to determine the selected item in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application displays the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1587 1252 1612"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel	Specifies the name of a callback function to be called to determine whether the application


Attributes	Description
(getShowLabel callback)	<p>displays the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1247 367"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 730 1198 753"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1115 1159 1138"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1524 1008 1547"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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</p>

Attributes	Description
	<p>MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 285 816 310"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 680 1299 989"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:ex="http://www.example.com"> <ribbon> <tabs> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier for an image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1428 1019 1453"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier of a built-in image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 993 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="Bold" /></pre> <p data-bbox="456 296 1433 380">This specifies a custom button that uses the built-in image with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 407 1401 457">Specifies the identifier of a built-in control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 468 1438 541">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 data-bbox="456 552 1005 573">For example, consider the following XML fragment:</p> <pre data-bbox="509 604 1273 674" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 730 1425 781">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 791 1433 842">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p data-bbox="456 861 1417 911">Specifies the qualified identifier of a control that this control is to be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 921 1438 995">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 data-bbox="456 1005 1005 1026">For example, consider the following XML fragment:</p> <pre data-bbox="509 1058 1286 1127" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1184 1430 1234">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 1245 1425 1295">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p data-bbox="456 1323 1417 1373">Specifies the identifier of a built-in control that this control is to be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1383 1438 1457">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 data-bbox="456 1467 1005 1488">For example, consider the following XML fragment:</p> <pre data-bbox="509 1520 1286 1589" style="background-color: #f0f0f0; padding: 5px;"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1646 1450 1696">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1707 1433 1757">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ	<p data-bbox="456 1776 1433 1797">Specifies the qualified identifier of a control that this control is to be inserted before. If the</p>

Attributes	Description
(qualified identifier of control to insert before)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1300 443"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 911 1198 934"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1213 1068 1236"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
onAction (onAction callback)	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1514 1300 1537"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>

Attributes	Description
	<p>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 627 1208 674"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon. The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon. For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1163 980 1209"><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays icons on its selection items. If this attribute is omitted, the items' icons SHOULD be shown by default. For example, consider the following XML fragment:</p> <pre data-bbox="509 1436 1289 1577"><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 that does not show any icons on its selection items. 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 displays labels on its selection items. If this attribute is omitted, the item's labels SHOULD be shown by default. For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1287 363"><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="457 417 1406 478">This specifies a gallery control that does not show any labels on its selection items. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
showLabel (show label)	<p data-bbox="457 501 1430 621">Specifies whether this control displays its label. The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 646 1159 695"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="457 747 1406 833">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
sizeString (size string)	<p data-bbox="457 854 1422 1003">Specifies a string whose size is 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 1029 1159 1052"><editBox id="editBox" sizeString="XXXXXXXXXXXX" /></pre> <p data-bbox="457 1104 1352 1215">This specifies an edit box control that is wide enough to display the string "XXXXXXXXXXXX". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
supertip (supertip)	<p data-bbox="457 1239 1398 1358">Specifies a string to be shown as the supertip of the control. The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown. For example, consider a control with a supertip, as follows:</p>  <p data-bbox="457 1753 1003 1778">This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1208 289"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p data-bbox="456 344 1349 394">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
tag (tag)	<p data-bbox="456 422 1393 493">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 data-bbox="456 506 1393 562">If this attribute is omitted, the control's tag value SHOULD default to an empty string. For example, consider the following XML fragment:</p> <pre data-bbox="509 590 1105 638"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p data-bbox="456 695 1279 743">This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p data-bbox="456 751 1349 800">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
visible (control visibility)	<p data-bbox="456 829 911 852">Specifies the visibility state of the control.</p> <p data-bbox="456 863 1365 911">The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control SHOULD default to being visible.</p> <p data-bbox="456 921 1003 945">For example, consider the following XML fragment:</p> <pre data-bbox="509 974 1003 997"><tab idMso="TabHome" visible="false" /></pre> <p data-bbox="456 1052 1260 1075">In this example, the built-in tab with an identifier of "TabHome" is hidden.</p> <p data-bbox="456 1085 1406 1108">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:

```
<xsd:complexType name="CT_DropDownRegular">
  <xsd:complexContent>
    <xsd:extension base="CT_Control">
      <xsd:sequence>
        <xsd:element name="item" type="CT_Item" minOccurs="0" maxOccurs="1000"/>
        <xsd:element name="button" type="CT_ButtonRegular" minOccurs="0" maxOccurs="16"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="AG_Action"/>
      <xsd:attributeGroup ref="AG_Enabled"/>

      <xsd:attributeGroup ref="AG_Image"/>
      <xsd:attributeGroup ref="AG_DropDownAttributes"/>
      <xsd:attribute name="getSelectedItemID" type="ST_Delegate" use="optional"/>
      <xsd:attribute name="getSelectedItemIndex" type="ST_Delegate" use="optional"/>
      <xsd:attribute name="showItemLabel" type="xsd:boolean" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



2.2.18 dynamicMenu (Unsize Dynamic Menu)


This element specifies a dynamic menu control that, because of its location, cannot have its anchor size changed. The **size** attribute is not present. It otherwise behaves identically to the regular **dynamicMenu** element, as specified in section [2.2.19](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called when the application needs to determine the contents of the control.</p> <p>For example, consider a dynamic menu control, as follows:</p>


Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 464 1094 510"><dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" /></pre> <p>The GetMenuContent callback function is called when the menu is dropped, and in this case would return a string with the following XML:</p> <pre data-bbox="509 642 1300 783"><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>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getDescription (getDescription callback)	<p>Specifies the name of a callback function to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1087 1276 1113"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1472 1162 1497"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>



Attributes	Description
	<pre data-bbox="509 218 1122 243"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="456 296 1390 348">In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p data-bbox="456 432 1451 485">Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p data-bbox="456 495 1349 548">The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 558 1003 590">For example, consider the following XML fragment:</p> <pre data-bbox="509 600 1146 625"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 678 1390 730">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 741 1382 793">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p data-bbox="456 810 1435 842">Specifies the name of a callback function to be called to determine the label of this control.</p> <p data-bbox="456 852 1427 905">The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 915 1003 947">For example, consider the following XML fragment:</p> <pre data-bbox="509 957 1122 982"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 1035 1451 1087">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 1098 1382 1150">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p data-bbox="456 1167 1390 1220">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 1230 1451 1314">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 data-bbox="456 1325 1003 1356">For example, consider the following XML fragment:</p> <pre data-bbox="509 1367 1224 1392"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="456 1444 1430 1497">In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p data-bbox="456 1507 1382 1560">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p data-bbox="456 1577 1435 1629">Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p data-bbox="456 1640 1357 1692">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1703 1003 1734">For example, consider the following XML fragment:</p>


Attributes	Description
	<pre data-bbox="509 218 1248 241" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="456 296 1333 348">In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p data-bbox="456 357 1385 409">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p data-bbox="456 434 1433 487">Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</p> <p data-bbox="456 493 1430 546">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 552 1008 575">For example, consider the following XML fragment:</p> <pre data-bbox="509 602 1248 625" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="456 680 1442 732">In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p data-bbox="456 741 1385 793">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p data-bbox="456 816 1385 869">Specifies the name of a callback function to be called to determine the supertip of this control.</p> <p data-bbox="456 875 1398 928">The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown.</p> <p data-bbox="456 934 1008 957">For example, consider the following XML fragment:</p> <pre data-bbox="509 984 1198 1008" style="text-align: center;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="456 1062 1417 1115">In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p data-bbox="456 1123 1385 1176">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p data-bbox="456 1201 1446 1253">Specifies the name of a callback function to be called to determine the visibility state of this control.</p> <p data-bbox="456 1260 1356 1312">The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible.</p> <p data-bbox="456 1318 1008 1341">For example, consider the following XML fragment:</p> <pre data-bbox="509 1369 1159 1392" style="text-align: center;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 1446 1450 1499">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 1507 1385 1560">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 1583 1408 1661">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 1667 1424 1719">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1726 1008 1749">For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 241" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 296 1390 380">This specifies a custom button control with an identifier of "MyButton". The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 407 1422 558">Specifies the identifier of a built-in control. The contents of this attribute are application-defined. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 585 816 609" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 663 1433 747">This creates a clone of the control with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 770 1433 951">Specifies a qualified identifier for a control. The idQ attribute can be used to reference controls or containers created by other Custom UI documents. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 978 1299 1287" style="background-color: #f0f0f0; padding: 5px;"><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 data-bbox="456 1341 1430 1505">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab. The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 1526 1442 1707">Specifies the relationship identifier 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. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1728 1019 1751" style="background-color: #f0f0f0; padding: 5px;"><button id="button" image="ForestPic" /></pre>

Attributes	Description
	<p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p>Specifies the identifier of a built-in image that is 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 541 997 569"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that uses the built-in image with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 926 1273 995"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1381 1289 1451"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1289 289"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 344 1443 394">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 405 1435 455">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p data-bbox="456 478 1443 529">Specifies the qualified identifier of a control that this control SHOULD be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 539 1443 617">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 data-bbox="456 627 1005 653">For example, consider the following XML fragment:</p> <pre data-bbox="509 680 1300 751"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 806 1443 856">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 867 1435 917">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
invalidateContentOnDrop (invalidate content on drop)	<p data-bbox="456 936 1435 987">Specifies whether this control SHOULD invalidate its contents and re-query for them when the user opens its drop-down menu.</p> <p data-bbox="456 997 1114 1022">If this attribute is omitted, its value SHOULD default to false.</p> <p data-bbox="456 1033 1005 1058">For example, consider the following XML fragment:</p> <pre data-bbox="509 1085 1260 1157"><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p data-bbox="456 1211 1435 1289">In this example, this combo box clears out its items and re-calls the GetComboBoxItemCount and GetComboBoxItemLabel callback functions to populate its contents each time the user opens it.</p> <p data-bbox="456 1299 1406 1325">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
keytip (keytip)	<p data-bbox="456 1344 1211 1369">Specifies a string to be used as the suggested KeyTip for this control.</p> <p data-bbox="456 1379 1349 1430">The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 1440 1089 1465">For example, consider a button with KeyTip 'K', as follows:</p>  <p data-bbox="456 1572 1005 1598">This is specified using the following XML fragment:</p> <pre data-bbox="509 1625 1198 1650"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p data-bbox="456 1705 1354 1755">The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p data-bbox="456 1776 1073 1801">Specifies a string to be used as the label for this control.</p>

Attributes	Description
	<p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1068 331"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1045 1208 1094"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1612 980 1661"><button id="button" showImage="false" label="Button with no icon" /></pre> <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>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1159 401"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1129 1208 1199"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1535 1105 1583"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
visible (control)	<p>Specifies the visibility state of the control.</p>

Attributes	Description
visibility)	<p>The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control SHOULD default to being visible.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1008 331" style="text-align: center;"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

2.2.19 dynamicMenu (Dynamic Menu)

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

For example, consider a dynamic menu control, as follows:



Figure 10: A dynamic menu control

This is 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>
```

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called when the application needs to determine the contents of the control.</p> <p>For example, consider a dynamic menu control, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><dynamicMenu id="dynamic" label="Dynamic Menu" getContent="GetMenuContent" /></pre> <p>The GetMenuContent callback function is called when the menu is dropped, and in this</p>


Attributes	Description
	<p>case would return a string with the following XML:</p> <pre data-bbox="509 254 1300 394"> <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>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getDescription (getDescription callback)</p>	<p>Specifies the name of a callback function to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 730 1273 753"> <button id="button" getDescription="GetButtonDescription" /> </pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1115 1159 1138"> <button id="button" getEnabled="IsButtonEnabled" /> </pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1472 1122 1495"> <button id="button" getImage="GetButtonImage" /> </pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p>


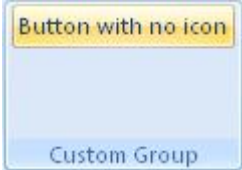
Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1146 279"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 611 1122 636"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1020 1224 1045"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1461 1248 1486"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute</p>



Attributes	Description
	<p>is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1248 310"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSize (getSize callback)</p>	<p>Specifies the name of a callback function to be called to determine the size of this control. The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size. For example, consider the following XML fragment:</p> <pre data-bbox="509 646 1094 672"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to be called to determine the supertip of this control. The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown. For example, consider the following XML fragment:</p> <pre data-bbox="509 1031 1198 1056"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to be called to determine the visibility state of this control. The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible. For example, consider the following XML fragment:</p> <pre data-bbox="509 1415 1159 1440"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 241" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 296 1390 380">This specifies a custom button control with an identifier of "MyButton". The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p data-bbox="201 407 391 457">idMso (built-in control identifier)</p>	<p data-bbox="456 407 1422 558">Specifies the identifier of a built-in control. The contents of this attribute are application-defined. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 585 816 609" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 663 1422 747">This creates a clone of the control with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 770 391 821">idQ (qualified control identifier)</p>	<p data-bbox="456 770 1430 951">Specifies a qualified identifier for a control. The idQ attribute can be used to reference controls or containers created by other Custom UI documents. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 978 1299 1287" style="background-color: #f0f0f0; padding: 5px;"><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 data-bbox="456 1344 1430 1503">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab. The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 1530 383 1581">image (custom image identifier)</p>	<p data-bbox="456 1530 1438 1703">Specifies the relationship identifier 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. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1730 1019 1753" style="background-color: #f0f0f0; padding: 5px;"><button id="button" image="ForestPic" /></pre>

Attributes	Description
	<p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 541 997 569"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 951 1273 1024"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1409 1289 1482"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1284 323"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 716 1284 785"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>invalidateContentOnDrop (invalidate content on drop)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1121 1260 1190"><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /></pre> <p>In this example, this combo box clears out its items and re-calls the GetComboBoxItemCount and GetComboBoxItemLabel callback functions to populate its contents each time the user opens it.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1661 1195 1682"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>

Attributes	Description
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1068 380"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1087 1208 1136"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1682 980 1730"><button id="button" showImage="false" label="Button with no icon" /></pre>

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 size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre data-bbox="505 457 1159 506"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="505 1024 1045 1052"><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="505 1745 1208 1791"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip"</pre>

Attributes	Description
	<pre>supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For 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 is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

2.2.20 editBox (Edit Box)

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

For example, consider an edit box control, as follows:

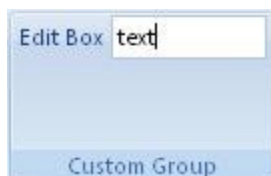


Figure 11: An edit box control

This is specified using the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the attributes of this element.


Attributes	Description
enabled (Enabled State)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p>


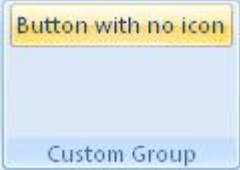
Attributes	Description
	<pre data-bbox="509 218 1146 243" style="text-align: center;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 296 1390 348">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 359 1385 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p data-bbox="456 432 1435 516">Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 527 1005 552">For example, consider the following XML fragment:</p> <pre data-bbox="509 579 1122 604" style="text-align: center;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 657 1451 709">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 720 1385 772">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p data-bbox="456 789 1395 842">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 852 1450 936">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 data-bbox="456 947 1005 972">For example, consider the following XML fragment:</p> <pre data-bbox="509 999 1224 1024" style="text-align: center;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="456 1077 1429 1129">In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p data-bbox="456 1140 1385 1192">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p data-bbox="456 1199 1435 1251">Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p data-bbox="456 1262 1357 1325">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1335 1005 1360">For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 1248 1413" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="456 1465 1333 1518">In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p data-bbox="456 1528 1385 1581">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p data-bbox="456 1577 1435 1640">Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</p> <p data-bbox="456 1650 1430 1713">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 1724 1005 1749">For example, consider the following XML fragment:</p>


Attributes	Description
	<pre data-bbox="509 218 1248 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p data-bbox="456 296 1442 348">In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p data-bbox="456 359 1385 407">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p data-bbox="456 432 1385 485">Specifies the name of a callback function to be called to determine the supertip of this control.</p> <p data-bbox="456 495 1398 548">The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown.</p> <p data-bbox="456 558 1005 579">For example, consider the following XML fragment:</p> <pre data-bbox="509 602 1198 623" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="456 680 1417 732">In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p data-bbox="456 743 1385 791">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getText (getText callback)	<p data-bbox="456 814 1450 867">Specifies the name of a callback function to be called to determine the text that SHOULD be displayed in the control.</p> <p data-bbox="456 877 1005 898">For example, consider the following XML fragment:</p> <pre data-bbox="509 926 1133 947" style="background-color: #f0f0f0; padding: 5px;"><editBox id="editBox" getText="GetEditBoxText" /></pre> <p data-bbox="456 1003 1450 1056">In this example, the GetEditBoxText callback function is called when the application needs to determine the text to display in the control.</p> <p data-bbox="456 1066 1385 1115">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p data-bbox="456 1140 1450 1192">Specifies the name of a callback function to be called to determine the visibility state of this control.</p> <p data-bbox="456 1203 1356 1255">The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible.</p> <p data-bbox="456 1266 1005 1287">For example, consider the following XML fragment:</p> <pre data-bbox="509 1314 1159 1335" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 1392 1450 1444">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 1455 1385 1503">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 1524 1409 1602">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 1612 1422 1665">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1675 1005 1696">For example, consider the following XML fragment:</p> <pre data-bbox="509 1724 1008 1745" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre>

Attributes	Description
	<p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 489 816 512" style="margin-left: 40px;"> <control idMso="Bold" /> </pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 884 1299 1192" style="margin-left: 40px;"> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1631 1019 1654" style="margin-left: 40px;"> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>

Attributes	Description
imageMso (built-in image identifier)	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 415 997 436"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 825 1273 894"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1283 1289 1352"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1738 1289 1787"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ...</pre>

Attributes	Description
	<p style="text-align: center;"></tab></p> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre style="text-align: center;"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>maxLength (maximum input string length)</p>	<p>Specifies an integer to be used as the maximum length of a string that can be entered into the control.</p> <p>If the maxLength attribute is omitted, the length of the input string SHOULD NOT be limited except by application-specific constraints.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 241"><editBox id="editBox" maxLength="10" /></pre> <p data-bbox="456 296 1369 348">This specifies an edit box control that can only accept strings up to 10 characters in length.</p> <p data-bbox="456 357 1433 409">The possible values for this attribute are defined by the ST_StringLength simple type, as specified in section 2.3.12.</p>
<p data-bbox="201 432 337 510">onChange (onChange callback)</p>	<p data-bbox="456 432 1438 485">Specifies the name of a callback function to be called when the text in the control has been changed by the user.</p> <p data-bbox="456 493 1005 516">For example, consider the following XML fragment:</p> <pre data-bbox="509 543 1198 567"><editBox id="editBox" onChange="EditBoxTextChanged" /></pre> <p data-bbox="456 621 1398 674">This specifies an edit box control that calls the EditBoxTextChanged callback function when the user inputs a text string.</p> <p data-bbox="456 682 1385 735">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 758 326 810">screentip (screentip)</p>	<p data-bbox="456 758 1138 781">Specifies a string to be shown as the screentip for this control.</p> <p data-bbox="456 789 1446 867">The screentip and getScreentip 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="456 875 1099 898">For example, consider a button with a screentip, as follows:</p>  <p data-bbox="456 1251 1002 1274">This is specified using the following XML fragment:</p> <pre data-bbox="509 1302 1208 1350"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 1404 1349 1457">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1482 420 1535">showImage (show image)</p>	<p data-bbox="456 1482 959 1505">Specifies whether this control displays an icon.</p> <p data-bbox="456 1514 1357 1566">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1575 1239 1598">For example, consider a button that does not display an icon, as follows:</p>  <p data-bbox="456 1787 1002 1810">This is specified using the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 980 264"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="457 317 1406 344">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 369 410 422">showLabel (Show Label)</p>	<p data-bbox="457 369 1430 489">Specifies whether this control displays its label. The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 516 1159 562"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="457 615 1406 701">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 726 386 779">sizeString (size string)</p>	<p data-bbox="457 726 1414 873">Specifies a string whose size is 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 900 1159 926"><editBox id="editBox" sizeString="XXXXXXXXXXXXXXXX" /></pre> <p data-bbox="457 978 1349 1083">This specifies an edit box control that is wide enough to display the string "XXXXXXXXXXXXXXXX". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1104 418 1136">supertip (supertip)</p>	<p data-bbox="457 1104 1398 1230">Specifies a string to be shown as the supertip of the control. The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown. For example, consider a control with a supertip, as follows:</p>  <p data-bbox="457 1623 1000 1650">This is specified using the following XML fragment:</p> <pre data-bbox="509 1677 1208 1751"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p data-bbox="457 1797 1349 1824">The possible values for this attribute are defined by the ST_String simple type, as</p>

Attributes	Description
	specified in section 2.3.11.
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. For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1105 470"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 806 1008 829"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_EditBox">
  <xsd:complexContent>
    <xsd:extension base="CT_Control">
      <xsd:attributeGroup ref="AG_Enabled"/>
      <xsd:attributeGroup ref="AG_Image"/>
      <xsd:attribute name="maxLength" type="ST_StringLength" use="optional"/>
      <xsd:attribute name="getText" type="ST_Delegate" use="optional"/>
      <xsd:attribute name="onChange" type="ST_Delegate" use="optional"/>
      <xsd:attribute name="sizeString" type="ST_String" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd: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 following its selection items.

For example, consider a gallery control that shows a selection of pictures, as follows:

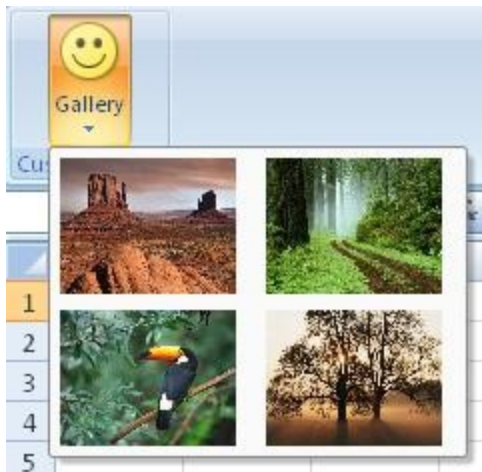


Figure 12: A gallery control

This is 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>
```

The following table summarizes the elements that are parents of this element.

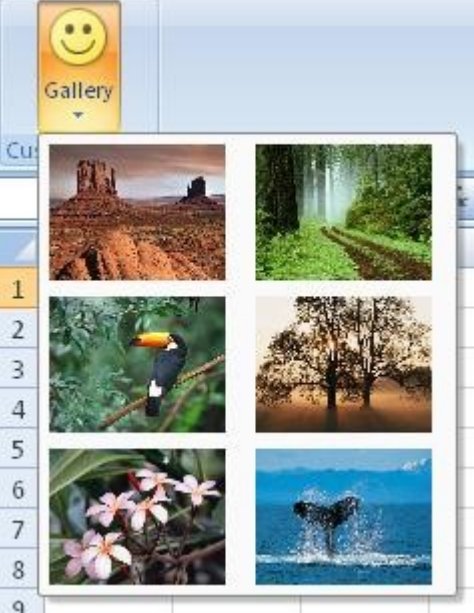

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
button (Unsize Button)	2.2.3
item (Selection Item)	2.2.24

The following table summarizes the attributes of this element.

Attributes	Description
columns (column count)	Specifies the number of columns that the gallery's items SHOULD be arranged into. If the columns attribute is omitted, the application SHOULD choose the number of columns automatically based on the number of items. For example, consider a gallery control with six items arranged into two columns, as follows:

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 877 1133 1094"> <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>The possible values for this attribute are defined by the ST_GalleryRowColumnCount simple type, as specified in section 2.3.4.</p>
<p>description (description)</p>	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1541 1263 1612"> <button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /> </pre> <p>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
<p>enabled (enabled state)</p>	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is</p>

Attributes	Description
	<p>specified, the control SHOULD default to being enabled.</p> <p>This attribute cannot be used to enable a built-in control that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1300 369"><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 705 1273 730"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1087 1159 1113"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1444 1122 1470"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemCount (getItemCount callback)</p>	<p>Specifies the name of a callback function to 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 that are specified as child elements. If no such items are specified, the control SHOULD be empty.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1260 241" style="background-color: #f0f0f0; padding: 5px;"><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p data-bbox="456 296 1446 348">In this example, the GetGalleryItemCount callback function is called when the application needs to determine the number of items in the gallery.</p> <p data-bbox="456 357 1382 409">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemHeight (getItemHeight callback)	<p data-bbox="456 432 1357 485">Specifies the name of a callback function to be called to determine the height of the selection items in this control.</p> <p data-bbox="456 493 1442 546">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 data-bbox="456 554 1414 606">The getItemHeight and getItemWidth attributes are mutually required. If only one of the attributes is specified, its value is ignored.</p> <p data-bbox="456 615 1003 638">For example, consider the following XML fragment:</p> <pre data-bbox="509 661 1219 735" style="background-color: #f0f0f0; padding: 5px;"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p data-bbox="456 789 1333 842">In this example, the GetGalleryItemHeight callback function is called when the application needs to determine the height of the items in the gallery.</p> <p data-bbox="456 850 1382 903">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemID (getItemID callback)	<p data-bbox="456 924 1365 976">Specifies the name of a callback function to be called to determine the identifier of a specific dynamically-created selection item, identified by index.</p> <p data-bbox="456 984 1365 1037">If this attribute is omitted, dynamically-created selection items SHOULD have empty identifiers.</p> <p data-bbox="456 1045 1003 1068">For example, consider the following XML fragment:</p> <pre data-bbox="509 1092 1219 1144" style="background-color: #f0f0f0; padding: 5px;"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p data-bbox="456 1199 1406 1251">In this example, the GetGalleryItemID callback function is called when the application needs to determine the identifier of a selection item.</p> <p data-bbox="456 1260 1382 1312">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemImage (getItemImage callback)	<p data-bbox="456 1333 1398 1386">Specifies the name of a callback function to be called to determine the icon of a specific dynamically-created selection item, identified by index.</p> <p data-bbox="456 1394 1438 1417">If this attribute is omitted, dynamically-created selection items SHOULD NOT display icons.</p> <p data-bbox="456 1425 1003 1449">For example, consider the following XML fragment:</p> <pre data-bbox="509 1472 1219 1524" style="background-color: #f0f0f0; padding: 5px;"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p data-bbox="456 1579 1333 1631">In this example, the GetGalleryItemImage callback function is called when the application needs to determine the icon of a selection item.</p> <p data-bbox="456 1640 1382 1692">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemLabel (getItemLabel callback)	<p data-bbox="456 1717 1406 1770">Specifies the name of a callback function to be called to determine the label of a specific dynamically-created selection item, identified by index.</p> <p data-bbox="456 1778 1446 1801">If this attribute is omitted, dynamically-created selection items SHOULD NOT display labels.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1224 300"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>In this example, the GetGalleryItemLabel callback function is called when the application needs to determine the label of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemScreentip (getItemScreentip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 663 1224 709"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>In this example, the GetGalleryItemScreentip callback function is called when the application needs to determine the screentip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemSupertip (getItemSupertip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1073 1224 1119"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>In this example, the GetGalleryItemSupertip callback function is called when the application needs to determine the supertip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getItemWidth (getItemWidth callback)</p>	<p>Specifies the name of a callback function to be called to determine the width of the selection items in this control.</p> <p>The itemWidth and getItemWidth 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 is ignored.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1539 1224 1612"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>In this example, the GetGalleryItemWidth callback function is called when the application needs to determine the width of the items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of</p>

Attributes	Description
(getKeytip callback)	<p>this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1146 369"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 701 1122 726"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1110 1224 1136"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSelectedItemI D (getSelectedItemI D callback)	<p>Specifies the name of a callback function to be called to determine the identifier of the item that is selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1497 1224 1566"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p>In this example, the GetGallerySelectedItemID callback function is called when the application needs to determine the selected item in the gallery. In this example the callback function returns one of the identifiers returned by the GetItemID callback function.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSelectedItemI	<p>Specifies the name of a callback function to be called to determine the index of the item to</p>



Attributes	Description
ndex (getSelectedItemIndex callback)	<p>be selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 346 1224 394"><gallery id="gallery" getItemCount="GetGalleryItemCount" getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p>In this example, the GetGallerySelectedItemIndex callback function is called when the application needs to determine the selected item in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 814 1252 835"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p>Specifies the name of a callback function to 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 is "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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1255 1252 1276"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSize (getSize callback)	<p>Specifies the name of a callback function to be called to determine the size of this control.</p> <p>The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1612 1097 1633"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>


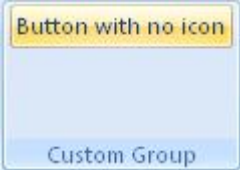
Attributes	Description
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to be called to determine the supertip of this control.</p> <p>The getSupertip and supertip attributes are mutually exclusive. If neither attribute is specified, no supertip for this control SHOULD be shown.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1198 405"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 764 1159 789"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1171 1008 1197"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1541 818 1566"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p>Specifies a qualified identifier for a control.</p> <p>The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p>

Attributes	Description
	<p>The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1299 621"> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1062 1019 1083"> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1478 997 1499"> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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</p>



Attributes	Description
	<p>appended to the existing set of controls, in the order they are defined in the XML. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1273 359"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1289 816"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1205 1289 1278"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1667 1305 1740"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the</p>

Attributes	Description
	<p>custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>invalidateContentOnDrop (invalidate content on drop)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 457 1260 527"> <comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel" invalidateContentOnDrop="true" /> </pre> <p>In this example, this combo box SHOULD clear out its items and re-call the GetComboBoxItemCount and GetComboBoxItemLabel callback functions to populate its contents each time the user opens it.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p>itemHeight (selection item height)</p>	<p>Specifies 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 itemHeight and itemWidth attributes are mutually required. If only one of the attributes is specified, its value is ignored.</p> <p>For example, consider a gallery control with 68 pixel tall items. This is specified using the following XML fragment:</p> <pre data-bbox="509 947 1187 1115"> <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>The possible values for this attribute are defined by the ST_GalleryItemWidthHeight simple type, as specified in section 2.3.3.</p>
<p>itemWidth (selection item width)</p>	<p>Specifies the width of the selection items in this control.</p> <p>The itemWidth and getItemWidth 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 itemHeight and itemWidth attributes are mutually required. If only one of the attributes is specified, its value is ignored.</p> <p>For example, consider a gallery control with 88 pixel wide items. This is specified using the following XML fragment:</p> <pre data-bbox="509 1478 1187 1646"> <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>The possible values for this attribute are defined by the ST_GalleryItemWidthHeight simple type, as specified in section 2.3.3.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p>

Attributes	Description
	<p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 443 1198 468"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1068 768"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user. For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1300 1066"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>In this example, the button calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>rows (row count)</p>	<p>Specifies the number of rows that the gallery's items are arranged into. If the rows attribute is omitted, the application SHOULD choose the number of rows automatically based on the number of items. For example, consider a gallery control with six items arranged into two rows, as follows:</p> 

Attributes	Description
	<p>This is specified using the following XML fragment:</p> <pre data-bbox="509 254 1094 470"> <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>The possible values for this attribute are defined by the ST_GalleryRowColumnCount simple type, as specified in section 2.3.4.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1146 1208 1192"> <button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /> </pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1738 980 1764"> <button id="button" showImage="false" </pre>

Attributes	Description
	<p style="text-align: center;"><code>label="Button with no icon" /></code></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 displays icons on its selection items. If this attribute is omitted, the items' icons SHOULD be shown by default. For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1289 625"><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 that does not show any icons on its selection items. 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 displays labels on its selection items. If this attribute is omitted, the item's labels SHOULD be shown by default. For example, consider the following XML fragment:</p> <pre data-bbox="509 877 1289 1024"><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>In this example, the gallery control does not show any labels on its selection items. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p>showInRibbon (show in ribbon)</p>	<p>This attribute has no meaning and MUST NOT be used.</p>
<p>showLabel (show label)</p>	<p>Specifies whether this control SHOULD display its label. This attribute SHOULD have no effect if the size or getSIZE attributes specify that the control is "large". The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 1444 1159 1493"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools. 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. The size and getSIZE attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size. For example, consider a large button, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 432 1045 453"><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
sizeString (size string)	<p>Specifies a string whose size is 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 758 1159 779"><editBox id="editBox" sizeString="XXXXXXXXXXXXXXXX" /></pre> <p>This specifies an edit box control that is wide enough to display the string "XXXXXXXXXXXXXXXX".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1535 1208 1608"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>

Attributes	Description
	<p>If this attribute is omitted, the control's tag value SHOULD default to an empty string. For 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 is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

2.2.22 gallery (Unsize Gallery)

This element specifies a gallery which, because of its location, cannot have its size changed. The **size** attribute is not present. It otherwise behaves identically to the regular **gallery** element, as specified in section [2.2.21](#).

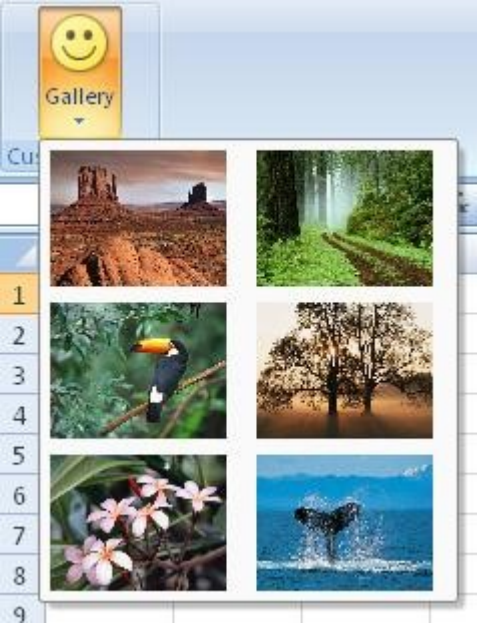

The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the child elements of this element.

Child Elements	Section
button (Unsize Button)	2.2.3
item (Selection Item)	2.2.24

The following table summarizes the attributes of this element.

Attributes	Description
<p>columns (column count)</p>	<p>Specifies the number of columns that the gallery's items are arranged into. If the columns attribute is omitted, the application SHOULD choose the number of columns automatically based on the number of items.</p> <p>For example, consider a gallery control with six items arranged into two columns, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1041 1133 1255"><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>The possible values for this attribute are defined by the ST_GalleryRowColumnCount simple type, as specified in section 2.3.4.</p>
<p>description (description)</p>	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1703 1263 1776"><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>The possible values for this attribute are defined by the <code>ST_LongString</code> simple type, as specified in section 2.3.8.</p>
<p>enabled (enabled state)</p>	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 516 1300 541"><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 873 1273 898"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1257 1159 1283"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1619 1122 1644"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>

Attributes	Description
getItemCount (getItemCount callback)	<p>Specifies the name of a callback function to 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 that are specified as child elements. If no such items are specified, the control SHOULD be empty.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1260 405"><gallery id="gallery" getItemCount="GetGalleryItemCount" /></pre> <p>In this example, the GetGalleryItemCount callback function is called when the application needs to determine the number of items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemHeight (getItemHeight callback)	<p>Specifies the name of a callback function to 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 is ignored.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 825 1219 894"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre> <p>In this example, the GetGalleryItemHeight callback function is called when the application needs to determine the height of the items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemID (getItemID callback)	<p>Specifies the name of a callback function to be called to determine the identifier of a specific dynamically-created selection item, identified by index.</p> <p>If this attribute is omitted, dynamically-created selection items SHOULD have empty identifiers.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1255 1219 1304"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetGalleryItemID" /></pre> <p>In this example, the GetGalleryItemID callback function is called when the application needs to determine the identifier of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemImage (getItemImage callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1640 1219 1688"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemImage="GetGalleryItemImage" /></pre> <p>In this example, the GetGalleryItemImage callback function is called when the application needs to determine the icon of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>

Attributes	Description
	specified in section 2.3.2.
getItemLabel (getItemLabel callback)	<p>Specifies the name of a callback function to 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 396 1224 443"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemLabel="GetGalleryItemLabel" /></pre> <p>In this example, the GetGalleryItemLabel callback function is called when the application needs to determine the label of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemScreentip (getItemScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 806 1224 852"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemScreentip="GetGalleryItemScreentip" /></pre> <p>In this example, the GetGalleryItemScreentip callback function is called when the application needs to determine the screentip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemSupertip (getItemSupertip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1205 1224 1251"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemSupertip="GetGalleryItemSupertip" /></pre> <p>In this example, the GetGalleryItemSupertip callback function is called when the application needs to determine the supertip of a selection item.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getItemWidth (getItemWidth callback)	<p>Specifies the name of a callback function to be called to determine the width of the selection items in this control.</p> <p>The itemWidth and getItemWidth 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 is ignored.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1677 1224 1749"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemHeight="GetGalleryItemHeight" getItemWidth="GetGalleryItemWidth" /></pre>


Attributes	Description
	<p>In this example, the GetGalleryItemWidth callback function is called when the application needs to determine the width of the items in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 506 1146 531"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1122 888"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1272 1224 1297"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSelectedItemID (getSelectedItemID callback)	<p>Specifies the name of a callback function to be called to determine the identifier of the item to be selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1661 1224 1730"><gallery id="gallery" getItemCount="GetGalleryItemCount" getItemID="GetItemID" getSelectedItemID="GetGallerySelectedItemID" /></pre> <p>In this example, the GetGallerySelectedItemID callback function is called when the</p>


Attributes	Description
	<p>application needs to determine the selected item in the gallery. The callback function returns one of the identifiers returned by the GetItemID callback function.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSelectedItemIndex (getSelectedItemIndex callback)</p>	<p>Specifies the name of a callback function to be called to determine the index of the item to be selected in this control.</p> <p>The getSelectedItemID and getSelectedItemIndex attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display a selected item.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 506 1224 554"><gallery id="gallery" getItemCount="GetGalleryItemCount" getSelectedItemIndex="GetGallerySelectedItemIndex" /></pre> <p>In this example, the GetGallerySelectedItemIndex callback function is called when the application needs to determine the selected item in the gallery.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 915 1252 940"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1299 1252 1325"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1682 1198 1707"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p>



Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1159 470" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getVisible="IsButtonVisible" /> </pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 856 1013 877" style="background-color: #f0f0f0; padding: 5px;"> <button id="MyButton" label="Button" /> </pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1226 818 1247" style="background-color: #f0f0f0; padding: 5px;"> <control idMso="Bold" /> </pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1617 1305 1785" style="background-color: #f0f0f0; padding: 5px;"> <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 data-bbox="509 205 716 321"></group> </tab> </tabs> </ribbon> </customUI></pre> <p data-bbox="459 373 1430 478">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="459 485 1425 537">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="459 562 1442 638">Specifies the relationship identifier 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 data-bbox="459 646 1438 699">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="459 707 1005 735">For example, consider the following XML fragment:</p> <pre data-bbox="509 758 1019 785"><button id="button" image="ForestPic" /></pre> <p data-bbox="459 837 1438 890">This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="459 898 1442 951">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="459 972 1446 1056">Specifies the identifier 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 data-bbox="459 1064 1438 1117">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="459 1125 1005 1152">For example, consider the following XML fragment:</p> <pre data-bbox="509 1176 995 1203"><button id="button" imageMso="Bold" /></pre> <p data-bbox="459 1255 1393 1308">This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p data-bbox="459 1316 1435 1369">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="459 1388 1446 1440">Specifies the identifier of a built-in control that this control SHOULD be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="459 1449 1438 1524">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 data-bbox="459 1533 1005 1560">For example, consider the following XML fragment:</p> <pre data-bbox="509 1583 1273 1656"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="459 1709 1425 1761">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="459 1770 1435 1797">The possible values for this attribute are defined by the ST_ID simple type, as specified in</p>


Attributes	Description
	section 2.3.5.
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1289 520"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1289 976"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1360 1289 1432"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1770 1260 1816"><comboBox id="comboBox" getItemCount="GetComboBoxItemCount" getItemLabel="GetComboBoxItemLabel"</pre>

Attributes	Description
	<pre>invalidateContentOnDrop="true" /></pre> <p>In this example, this combo box SHOULD clear out its items and re-call the GetComboBoxItemCount and GetComboBoxItemLabel callback functions to populate its contents each time the user opens it.</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 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 itemHeight and itemWidth attributes are mutually required. If only one of the attributes is specified, its value is ignored.</p> <p>For example, consider a gallery control with 68 pixel tall items. This is 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>The possible values for this attribute are defined by the ST_GalleryItemWidthHeight simple type, as specified in section 2.3.3.</p>
itemWidth (selection item width)	<p>Specifies the width of the selection items in this control.</p> <p>The itemWidth and getItemWidth 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 itemHeight and itemWidth attributes are mutually required. If only one of the attributes is specified, its value is ignored.</p> <p>For example, consider a gallery control with 88 pixel wide items. This is 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>The possible values for this attribute are defined by the ST_GalleryItemWidthHeight simple type, as specified in section 2.3.3.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1068 447"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 722 1300 747"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>In this example, the button calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>rows (row count)</p>	<p>Specifies the number of rows that the gallery's items are arranged into.</p> <p>If the rows attribute is omitted, the application SHOULD choose the number of rows automatically based on the number of items.</p> <p>For example, consider a gallery control with six items arranged into two rows, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1556 1094 1772"><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>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_GalleryRowColumnCount simple type, as specified in section 2.3.4.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 825 1209 871"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1360 980 1407"><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays icons on its selection items.</p> <p>If this attribute is omitted, the items' icons SHOULD be shown by default.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1665 1287 1787"><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" /></pre>

Attributes	Description
	<p data-bbox="509 201 639 222"></gallery></p> <p data-bbox="456 279 1406 338">This specifies a gallery control that does not show any icons on its selection items. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 363 394 415">showItemLabel (show item label)</p>	<p data-bbox="456 363 1179 422">Specifies whether this control displays labels on its selection items. For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1287 590"> <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="456 646 1406 705">In this example, the gallery control does not show any labels on its selection items. The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 730 404 783">showLabel (show label)</p>	<p data-bbox="456 730 1053 751">Specifies whether this control SHOULD display its label.</p> <p data-bbox="456 798 1430 846">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 856 1005 877">For example, consider the following XML fragment:</p> <pre data-bbox="509 909 1159 957"> <button id="button" label="Label" showLabel="false" imageMso="HappyFace" /> </pre> <p data-bbox="456 1010 1406 1058">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p data-bbox="456 1068 1406 1089">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 1119 386 1171">sizeString (size string)</p>	<p data-bbox="456 1119 1406 1167">Specifies a string whose size is used to determine the width of the text input area of this control.</p> <p data-bbox="456 1178 1406 1226">If this attribute is omitted, the application SHOULD determine the width of the text input area of the control automatically.</p> <p data-bbox="456 1236 1005 1257">For example, consider the following XML fragment:</p> <pre data-bbox="509 1289 1159 1310"> <editBox id="editBox" sizeString="XXXXXXXXXXXXXXXX" /> </pre> <p data-bbox="456 1367 1260 1415">This specifies an edit box control that is wide enough to display the string "XXXXXXXXXXXXXXXX".</p> <p data-bbox="456 1425 1349 1474">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1501 418 1522">supertip (supertip)</p>	<p data-bbox="456 1501 1114 1522">Specifies a string to be shown as the supertip of the control.</p> <p data-bbox="456 1533 1398 1581">The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown.</p> <p data-bbox="456 1591 1092 1612">For example, consider a control with a supertip, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 638 1208 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 1010 1105 1058"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1394 1008 1415"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_GalleryRegular">
  <xsd:complexContent>
    <xsd:extension base="CT_DropDownRegular">
      <xsd:attributeGroup ref="AG_Description"/>
      <xsd:attributeGroup ref="AG_DynamicContentAttributes"/>
      <xsd:attribute name="columns" type="ST_GalleryRowColumnCount" use="optional"/>
      <xsd:attribute name="rows" type="ST_GalleryRowColumnCount" use="optional"/>
      <xsd:attribute name="itemWidth" type="ST_GalleryItemWidthHeight" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

<xsd:attribute name="itemHeight" type="ST_GalleryItemWidthHeight" use="optional"/>
<xsd:attribute name="getItemWidth" type="ST_Delegate" use="optional"/>
<xsd:attribute name="getItemHeight" type="ST_Delegate" use="optional"/>
<xsd:attribute name="showItemLabel" type="xsd:boolean" use="optional"/>
</xsd:extension>
</xsd:complexContent>
</xsd: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**.

For example, consider a group with a single button, as follows:



Figure 13: A group with a single button

This is specified using the following XML fragment:

```

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

```

The following table summarizes the elements that are parents of this element.

Parent Elements
tab (section 2.2.39)

The following table summarizes the child elements of this element.

Child Elements	Section
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
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

Child Elements	Section
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


The following table summarizes the attributes of this element.



Attributes	Description
getImage (getImage callback)	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip	Specifies the name of a callback function to be called to determine the screentip of this

Attributes	Description
(getScreentip callback)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1224 394"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 758 1198 779"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1142 1159 1163"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1549 1008 1570"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier of a built-in control.</p> <p>The contents of this attribute are application-defined.</p>

Attributes	Description
	<p>The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 816 331"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 709 1299 1014"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1455 1019 1476"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 249 993 275"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 659 1273 730"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1121 1286 1192"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1575 1286 1646"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>

Attributes	Description
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 407 1300 478"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 947 1198 972"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1247 1068 1272"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 596 1208 642"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>supertip (supertip)</p>	<p>Specifies a string to be shown as the supertip of the control. The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown. For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1339 1208 1407"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. If this attribute is omitted, the control's tag value SHOULD default to an empty string. For example, consider the following XML fragment:</p> <pre data-bbox="509 1709 1110 1755"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre>

Attributes	Description
	<p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1008 506" style="text-align: center;"> <tab idMso="TabHome" visible="false" /> </pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```

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

```

2.2.24 item (Selection Item)

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

For example, consider a drop-down control with three selection items, as follows:

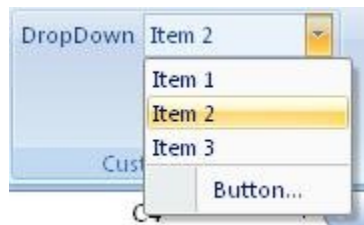


Figure 14: A drop-down control with selection items

This is 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>



```

The following table summarizes the elements that are parents of this element.

Parent Elements
comboBox (section 2.2.7); dropDown (section 2.2.17); gallery (section 2.2.21); gallery (section 2.2.22)

The following table summarizes the attributes of this element.

Attributes	Description
id (custom control identifier)	<p>Specifies the identifier for a custom control. All new custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
image (custom image identifier)	<p>Specifies the identification information for an image to 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 image, and imageMso attributes are mutually exclusive.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" image="ForestPic" /></pre> <p>In this example, the custom button has an icon that is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p>Specifies the identifier of a built-in image to 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 image, and imageMso attributes are mutually exclusive.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" imageMso="Bold" /></pre> <p>In this example, the custom button uses the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1068 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Custom Button" /></pre> <p data-bbox="456 296 1349 380">This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 407 326 457">screentip (screentip)</p>	<p data-bbox="456 407 1138 464">Specifies a string to be shown as the screentip for this control. For example, consider a button with a screentip, as follows:</p>  <p data-bbox="456 816 1000 842">This is specified using the following XML fragment:</p> <pre data-bbox="509 869 1208 915" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 968 1349 1024">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1043 418 1073">supertip (supertip)</p>	<p data-bbox="456 1043 1114 1100">Specifies a string to be shown as the supertip of the control. For example, consider a control with a supertip, as follows:</p>  <p data-bbox="456 1497 1000 1522">This is specified using the following XML fragment:</p> <pre data-bbox="509 1549 1208 1623" 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 data-bbox="456 1675 1349 1732">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>

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

```

<xsd:complexType name="CT_Item">
  <xsd:attribute name="id" type="ST_UniqueID" use="optional"/>
  <xsd:attribute name="label" type="ST_String" use="optional"/>
  <xsd:attribute name="image" type="ST Uri" use="optional"/>
  <xsd:attribute name="imageMso" type="ST_ID" use="optional"/>
  <xsd:attribute name="screentip" type="ST String" use="optional"/>
  <xsd:attribute name="supertip" type="ST_String" use="optional"/>
</xsd:complexType>

```

2.2.25 labelControl (Text Label)

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

For example, consider a label control, as follows:

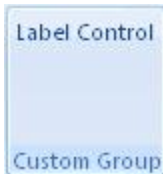


Figure 15: A label control

This is specified using the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the attributes of this element.


Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p>



Attributes	Description
	<pre data-bbox="509 218 1159 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="456 296 1393 348">In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p data-bbox="456 359 1385 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 432 334 506">getImage (getImage callback)</p>	<p data-bbox="456 432 1438 516">Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 527 1008 548">For example, consider the following XML fragment:</p> <pre data-bbox="509 579 1122 600" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="456 653 1393 705">In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p data-bbox="456 716 1385 768">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 789 334 863">getKeytip (getKeytip callback)</p>	<p data-bbox="456 789 1451 842">Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p data-bbox="456 852 1349 905">The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 915 1008 936">For example, consider the following XML fragment:</p> <pre data-bbox="509 968 1146 989" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 1041 1393 1094">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 1104 1385 1157">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1167 431 1220">getLabel (getLabel callback)</p>	<p data-bbox="456 1167 1438 1220">Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 1230 1008 1251">For example, consider the following XML fragment:</p> <pre data-bbox="509 1283 1122 1304" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 1356 1451 1409">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 1419 1385 1472">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1524 367 1598">getScreentip (getScreentip callback)</p>	<p data-bbox="456 1524 1393 1577">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 1587 1451 1640">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 data-bbox="456 1650 1008 1671">For example, consider the following XML fragment:</p> <pre data-bbox="509 1703 1224 1724" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getScreentip="GetButtonScreentip" /></pre>


Attributes	Description
	<p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 506 1248 531" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 890 1248 915" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1274 1198 1299" style="text-align: center;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1661 1159 1686" style="text-align: center;"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>

Attributes	Description
	specified in section 2.3.2.
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1008 470"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 816 818 837"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1209 1300 1518"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p>Specifies the relationship identifier 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</p>

Attributes	Description
	<p>containing file.</p> <p>The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1019 369"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 764 993 789"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1171 1273 1247"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1633 1289 1709"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p>

Attributes	Description
	The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 474 1289 541"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 930 1300 997"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1472 1198 1493"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1070 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Custom Button" /></pre> <p data-bbox="456 296 1349 380">This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 407 324 457">screentip (screentip)</p>	<p data-bbox="456 407 1446 516">Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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="456 527 1101 548">For example, consider a button with a screentip, as follows:</p>  <p data-bbox="456 905 1003 926">This is specified using the following XML fragment:</p> <pre data-bbox="509 953 1208 999" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 1087 1349 1136">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1163 422 1213">showImage (show image)</p>	<p data-bbox="456 1163 1357 1251">Specifies whether this control displays an icon. The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1262 1240 1283">For example, consider a button that does not display an icon, as follows:</p>  <p data-bbox="456 1472 1003 1493">This is specified using the following XML fragment:</p> <pre data-bbox="509 1520 980 1566" style="background-color: #f0f0f0; padding: 5px;"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="456 1661 1406 1682">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 1709 406 1759">showLabel (show label)</p>	<p data-bbox="456 1709 1052 1730">Specifies whether this control SHOULD display its label.</p> <p data-bbox="456 1740 1430 1787">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 247 1159 296"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1024 1208 1100"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1398 1110 1451"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 239" style="font-family: monospace;"><tab idMso="TabHome" visible="false" /></pre> <p data-bbox="456 296 1406 354">In this example, the built-in tab with an identifier of "TabHome" is hidden. 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:

```

<xsd:complexType name="CT_LabelControl">
  <xsd:complexContent>
    <xsd:restriction base="CT_Control">
      <xsd:attribute name="image" use="prohibited"/>
      <xsd:attribute name="imageMso" use="prohibited"/>
      <xsd:attribute name="getImage" use="prohibited"/>
      <xsd:attribute name="keytip" use="prohibited"/>

      <xsd:attribute name="getKeytip" use="prohibited"/>
      <xsd:attribute name="showImage" use="prohibited"/>
      <xsd:attribute name="getShowImage" use="prohibited"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

```

2.2.26 menu (Unsize Menu)

This element specifies a menu control that, because of its location, cannot have its size changed. The **size** attribute is not present. It otherwise behaves identically to the regular **menu** element, as specified in section [2.2.28](#).


The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); splitButton (section 2.2.38); splitButton (section 2.2.36)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.


Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 579 1260 646"><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 982 1299 1008"><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1346 1273 1371"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1728 1159 1753"><button id="button" getEnabled="IsButtonEnabled" /></pre>



Attributes	Description
	<p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1122 506" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1146 890" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getLabel (getLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1220 1122 1247" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getScreentip (getScreentip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1629 1224 1656" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>

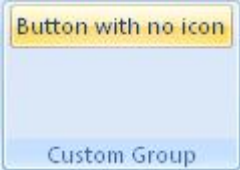

Attributes	Description
	specified in section 2.3.2.
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1248 447"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 806 1248 831"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1190 1198 1215"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1575 1159 1600"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control)	Specifies the identifier for a custom control. All custom controls MUST have unique

Attributes	Description
<p>identifier)</p>	<p>identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 369 1008 394"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 737 816 762"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1131 1300 1440"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1019 275" style="margin-left: 40px;"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 674 997 695" style="margin-left: 40px;"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1083 1273 1150" style="margin-left: 40px;"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1541 1289 1608" style="margin-left: 40px;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. If</p>

Attributes	Description
(identifier of built-in control to insert before)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1289 443"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 831 1300 900"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
itemSize (item size)	<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>For example, consider a menu control with large menu items, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1591 1312 1709"><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>The possible values for this attribute are defined by the ST_ItemSize simple type, as specified in section 2.3.6.</p>

Attributes	Description
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 485 1198 510"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 785 1070 810"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1520 1208 1570"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p>

Attributes	Description
	<p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 464 980 512"><button id="button" showImage="false" label="Button with no icon" /></pre> <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>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 793 1159 842"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1570 1208 1646"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1105 426"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 762 1008 787"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

2.2.27 menu (Menu with Title)

This element specifies a menu control that, because of its location, can optionally include a title string via the **title** or **getTitle** attributes. It otherwise behaves identically to the regular **menu** element, as specified in section [2.2.28](#).

For example, consider a menu control with a title, as follows:

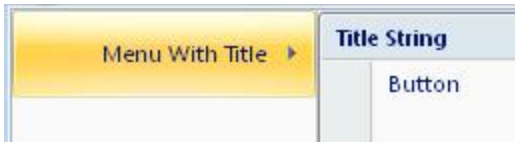


Figure 16: A menu control with title

This is specified with the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
menu (section 2.2.27); officeMenu (section 2.2.31); splitButton (section 2.2.37)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>



Attributes	Description
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1159 401" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getEnabled="IsButtonEnabled" /> </pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 737 1122 758" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getImage="GetButtonImage" /> </pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1121 1146 1142" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getKeytip="GetButtonKeytip" /> </pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1478 1122 1499" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getLabel="GetButtonLabel" /> </pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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</p>



Attributes	Description
	<p>no screentip at all.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1224 310"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 669 1248 695"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1054 1248 1079"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1438 1198 1463"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getTitle (getTitle callback)</p>	<p>Specifies the name of a callback function to be called to determine the title of this control.</p> <p>The title and getTitle attributes are mutually exclusive. If neither attribute is specified no title SHOULD be shown.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1795 1187 1820"><menu id="menu" label="Menu" getTitle="GetMenuTitle"></pre>


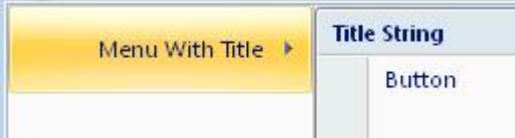
Attributes	Description
	<p>...</p> <pre data-bbox="509 222 597 247"></menu></pre> <p>In this example, the GetMenuTitle callback function is called when the application needs to determine the title of the menu.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 611 1159 636"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1020 1003 1045"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 815 1413"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1780 1299 1822"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"></pre>

Attributes	Description
	<pre data-bbox="509 201 1105 464"> 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 data-bbox="456 516 1430 617">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 627 1425 680">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 703 1442 781">Specifies the relationship identifier 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 data-bbox="456 789 1438 842">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 850 1005 875">For example, consider the following XML fragment:</p> <pre data-bbox="509 900 1018 926"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 978 1438 1031">This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 1039 1442 1092">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 1115 1446 1197">Specifies the identifier 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 data-bbox="456 1205 1438 1260">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1268 1005 1293">For example, consider the following XML fragment:</p> <pre data-bbox="509 1318 993 1344"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1396 1395 1449">This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p data-bbox="456 1457 1435 1509">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1535 1446 1587">Specifies the identifier of a built-in control that this control SHOULD be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1596 1438 1673">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 data-bbox="456 1682 1005 1707">For example, consider the following XML fragment:</p> <pre data-bbox="509 1732 1273 1774"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre>

Attributes	Description
	<p style="text-align: center;"></tab></p> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre style="text-align: center;"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>itemSize (item</p>	<p>Specifies the size of the child controls in this menu.</p>

Attributes	Description
size)	<p>If this attribute is omitted, the menu's child controls SHOULD default to the normal size. For example, consider a menu control with large menu items, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 667 1312 783"><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>The possible values for this attribute are defined by the ST_ItemSize simple type, as specified in section 2.3.6.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control. The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1192 1198 1220"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1493 1068 1520"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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>

Attributes	Description
	<p>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 632 1211 678"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1199 980 1245"><button id="button" showImage="false" label="Button with no icon" /></pre> <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>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1528 1159 1575"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 to be shown as the supertip of the control.</p> <p>The supertip and getSupertip attributes are mutually exclusive. If neither attribute is</p>

Attributes	Description
	<p>specified no supertip for this control SHOULD be shown. For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 705 1208 779"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. If this attribute is omitted, the control's tag value SHOULD default to an empty string. For example, consider the following XML fragment:</p> <pre data-bbox="509 1110 1105 1161"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function. The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>title (title)</p>	<p>Specifies a string to be displayed as the title of the control. The title and getTitle attributes are mutually exclusive. If neither attribute is specified, no title SHOULD be shown. For example, consider a menu control with a title, as follows:</p>  <p>This is specified with the following XML fragment:</p> <pre data-bbox="509 1673 1287 1745"><menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as</p>

Attributes	Description
	specified in section 2.3.11.
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>For example, consider the following XML fragment:</p> <pre data-bbox="509 396 1008 422" style="text-align: center;"> <tab idMso="TabHome" visible="false" /> </pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```

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

```

2.2.28 menu (Menu)

This element specifies a drop-menu control.

For example, consider a menu control, as follows:



Figure 17: A menu control

This is 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>

```


The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre>



Attributes	Description
	<p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1273 506" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getDescription="GetButtonDescription" /> </pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1159 890" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getEnabled="IsButtonEnabled" /> </pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1220 1122 1247" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getImage="GetButtonImage" /> </pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1604 1146 1631" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getKeytip="GetButtonKeytip" /> </pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>



Attributes	Description
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1122 380"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 764 1224 791"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1205 1248 1232"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p>Specifies the name of a callback function to 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 is "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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1650 1248 1677"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as</p>



Attributes	Description
	specified in section 2.3.2.
getSize (getSize callback)	<p>Specifies the name of a callback function to be called to determine the size of this control. The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 394 1094 422"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 779 1198 806"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1163 1159 1190"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1572 1008 1600"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier of a built-in control.</p>

Attributes	Description
	<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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 818 365" style="background-color: #f0f0f0; padding: 5px;"> <control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 737 1299 1050" style="background-color: #f0f0f0; padding: 5px;"> <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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1486 1019 1507" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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</p>

Attributes	Description
	<p>attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 285 997 310"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 695 1273 768"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1157 1289 1230"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1608 1289 1682"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in</p>

Attributes	Description
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>section 2.3.5.</p> <p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>itemSize (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>For example, consider a menu control with large menu items, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_ItemSize simple type, as specified in section 2.3.6.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" keytip="K" /></pre>

Attributes	Description
	<p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 422 1070 447"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1157 1208 1203"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1751 980 1776"><button id="button" showImage="false"</pre>

Attributes	Description
	<pre>label="Button with no icon" /></pre> <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 size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p> 

Attributes	Description
	<p>This is specified using the following XML fragment:</p> <pre data-bbox="509 254 1208 323"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 627 1105 674"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1010 1008 1031"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

For example, consider a dynamic menu control, as follows:



Figure 18: A dynamic menu control

This is 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 returns a string with the following XML:


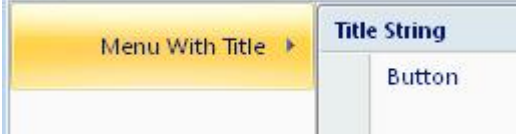
```
<menu xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <button id="button1" label="Button 1" />
  <button id="button2" label="Button 2" />
  <button id="button3" label="Button 3" />
</menu>
```

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
getTitle (getTitle callback)	<p>Specifies the name of a callback function to be called to determine the title of this control. The title and getTitle attributes are mutually exclusive. If neither attribute is specified no title SHOULD be shown.</p> <p>For example, consider the following XML fragment:</p> <pre><menu id="menu" label="Menu" getTitle="GetMenuTitle"> ... </menu></pre>

Attributes	Description
	<p>In this example, the GetMenuTitle callback function is to be called when the application needs to determine the title of the menu.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>itemSize (item size)</p>	<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>For example, consider a menu control with large menu items, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 863 1312 978"> <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>The possible values for this attribute are defined by the ST_ItemSize simple type, as specified in section 2.3.6.</p>
<p>title (title)</p>	<p>Specifies a string to 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>For example, consider a menu control with a title, as follows:</p>  <p>This is specified with the following XML fragment:</p> <pre data-bbox="509 1436 1284 1503"> <menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu> </pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>

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

```

<xsd:complexType name="CT_MenuRoot">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:group ref="EG_MenuControlsBase"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:group ref="EG_MenuOrSplitButtonRegular"/>
</xsd:choice>
</xsd:sequence>
<xsd:attributeGroup ref="AG_Title"/>
<xsd:attribute name="itemSize" type="ST_ItemSize" use="optional"/>
</xsd: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.

For example, consider a menu with a separator in between two of its items, as follows:



Figure 19: Menu control with separator

This is 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>

```

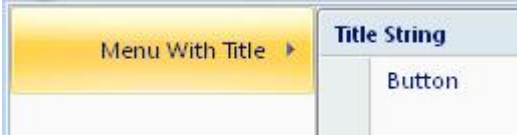
The following table summarizes the elements that are parents of this element.

Parent Elements
menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the attributes of this element.

Attributes	Description
getTitle (getTitle callback)	<p>Specifies the name of a callback function to be called to determine the title of this control. The title and getTitle attributes are mutually exclusive. If neither attribute is specified no title SHOULD be shown.</p> <p>For example, consider the following XML fragment:</p> <pre> <menu id="menu" label="Menu" getTitle="GetMenuTitle"> ... </menu> </pre> <p>In this example, the GetMenuTitle callback function is called when the application needs to determine the title of the menu.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control)	Specifies the identifier for a custom control. All custom controls MUST have unique

Attributes	Description
<p>identifier)</p>	<p>identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>The id and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 369 1008 394"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 764 1300 1073"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1514 1276 1583"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted after. If</p>

Attributes	Description
of control to insert after)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1289 443"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 831 1289 900"> <tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1287 1289 1356"> <tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
title (title)	<p>Specifies a string to 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>For example, consider a menu control with a title, as follows:</p> 

Attributes	Description
	<p>This is specified with the following XML fragment:</p> <pre data-bbox="509 254 1284 323"> <menu id="menu" label="Menu With Title" title="Title String"> <button id="button" label="Button" /> </menu> </pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>

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

```

<xsd:complexType name="CT_MenuSeparator">
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
  <xsd:attributeGroup ref="AG_Title"/>
</xsd: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.

For example, consider the following XML fragment:

```

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

```

This XML fragment specifies that the command with an identifier of "FileSave" on the Office Menu is hidden.

The following table summarizes the elements that are parents of this element.

Parent Elements
ribbon (section 2.2.33)

The following table summarizes the child elements of this element.

Child Elements	Section
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

Child Elements	Section
splitButton (Split Button with Title)	2.2.37
toggleButton (Unsize Toggle Button)	2.2.42

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

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

For example, consider the following controls on the document-specific quick access toolbar:



Figure 20: Controls on the quick access toolbar

This is specified using the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
ribbon (section 2.2.33)

The following table summarizes the child elements of this element.

Child Elements	Section
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:

```

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

```

2.2.33 ribbon (Ribbon)

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

The following table summarizes the elements that are parents of this element.

Parent Elements
customUI (section 2.2.14)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.

Attributes	Description
startFromScratch (start from scratch)	<p>Specifies that the application's built-in ribbon UI is 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 "false". For example, consider the following XML fragment:</p> <pre> <ribbon startFromScratch="true"> ... </ribbon> </pre> <p>In this example, the application's ribbon is put into start from scratch mode. 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:

```

<xsd:complexType name="CT_Ribbon">
  <xsd:all>
    <xsd:element name="officeMenu" type="CT_OfficeMenu" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="qat" type="CT_Qat" minOccurs="0" maxOccurs="1"/>
    <xsd:unique name="qatControls">
      <xsd:selector xpath="*/*/"/>
      <xsd:field xpath="@id"/>
    </xsd:unique>
  </xsd:all>
</xsd:complexType>

```

```

<xsd:element name="tabs" type="CT_Tabs" minOccurs="0" maxOccurs="1"/>
<xsd:element name="contextualTabs" type="CT_ContextualTabs" minOccurs="0" maxOccurs="1"/>
</xsd:all>
<xsd:attribute name="startFromScratch" type="xsd:boolean" use="optional"/>
</xsd: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.

For example, consider a vertical separator control between two buttons, as follows:



Figure 21: A vertical separator control

This is 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" />

```

The following table summarizes the elements that are parents of this element.

Parent Elements
documentControls (section 2.2.16); group (section 2.2.23); sharedControls (section 2.2.35)

The following table summarizes the attributes of this element.

Attributes	Description
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p>The id and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 247 1008 275"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 and idQ attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 646 1300 953"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1394 1276 1465"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1289 323"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 716 1289 785"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1167 1289 1236"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1577 1008 1600"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```

<xsd:complexType name="CT_Separator">
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
</xsd: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 among 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.

For example, consider a Ribbon Extensibility XML document that adds the two buttons to the shared section of the quick access toolbar:



Figure 22: Shared controls on the quick access toolbar

This is specified using the following XML fragment:

```

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

```

The following table summarizes the elements that are parents of this element.

Parent Elements
qat (section 2.2.32)

The following table summarizes the child elements of this element.

Child Elements	Section
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:

```

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

```

</xsd:complexType>

2.2.36 splitButton (Unsize Split Button)

This element specifies a split button control that, because of its location, cannot have its size changed. The **size** attribute is not present. It otherwise behaves identically to the regular **splitButton** element, as specified in section [2.2.38](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.


Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage)	<p>Specifies the name of a callback function to be called to determine the icon of this control.</p>


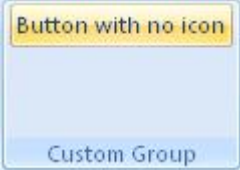
Attributes	Description
callback)	<p>The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 310 1122 338"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 695 1146 722"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control.</p> <p>The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1056 1122 1083"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1465 1224 1493"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p>


Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 247 1247 275"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 636 1247 663"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1020 1198 1047"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1404 1159 1432"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 239" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 296 1390 380">This specifies a custom button control with an identifier of "MyButton". The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p data-bbox="201 407 391 457">idMso (built-in control identifier)</p>	<p data-bbox="456 407 1422 558">Specifies the identifier of a built-in control. The contents of this attribute are application-defined. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 585 816 606" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 663 1433 747">This creates a clone of the control with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 774 391 825">idQ (qualified control identifier)</p>	<p data-bbox="456 774 1433 951">Specifies a qualified identifier for a control. The idQ attribute can be used to reference controls or containers created by other Custom UI documents. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 978 1300 1287" style="background-color: #f0f0f0; padding: 5px;"><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 data-bbox="456 1344 1433 1503">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab. The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 1530 383 1581">image (custom image identifier)</p>	<p data-bbox="456 1530 1433 1698">Specifies the relationship identifier 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. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1726 1019 1747" style="background-color: #f0f0f0; padding: 5px;"><button id="button" image="ForestPic" /></pre>

Attributes	Description
	<p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 541 997 567"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 951 1273 1024"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1409 1289 1482"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1284 323"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 709 1284 779"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1251 1195 1274"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1549 1068 1572"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD display the label of the control as the screentip or display</p>

Attributes	Description
	<p>no screentip at all. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 659 1211 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1230 980 1281"><button id="button" showImage="false" label="Button with no icon" /></pre> <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>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1596 1159 1646"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>

Attributes	Description
<p>supertip (supertip)</p>	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 772 1208 846"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1146 1105 1192"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1528 1008 1554"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_SplitButtonRegular">
<xsd:complexContent>
```

```

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

```

2.2.37 splitButton (Split Button with Title)

This element specifies a split button control that, because of its location, can optionally include a title string via the **title** or **getTitle** attributes. It otherwise behaves identically to the regular **splitButton** element, as specified in section [2.2.38](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the child elements of this element.

Child Elements	Section
button (Button Inside of a Split Button)	2.2.4
menu (Menu with Title)	2.2.27
toggleButton (Toggle Button Inside of a Split Button)	2.2.44

The following table summarizes the attributes of this element.


Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p>


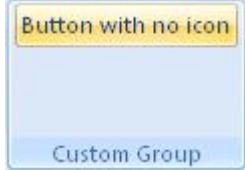
Attributes	Description
	<pre data-bbox="509 218 1159 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p data-bbox="456 296 1390 348">In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 432 334 506">getImage (getImage callback)</p>	<p data-bbox="456 432 1430 516">Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 527 1003 548">For example, consider the following XML fragment:</p> <pre data-bbox="509 579 1122 600" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getImage="GetButtonImage" /></pre> <p data-bbox="456 653 1390 705">In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p data-bbox="456 716 1382 768">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 789 334 863">getKeytip (getKeytip callback)</p>	<p data-bbox="456 789 1446 842">Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p data-bbox="456 852 1344 905">The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 915 1003 936">For example, consider the following XML fragment:</p> <pre data-bbox="509 968 1146 989" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 1041 1390 1094">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 1104 1382 1157">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1167 431 1220">getLabel (getLabel callback)</p>	<p data-bbox="456 1167 1430 1220">Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 1230 1003 1251">For example, consider the following XML fragment:</p> <pre data-bbox="509 1283 1122 1304" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 1356 1446 1409">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 1419 1382 1472">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p data-bbox="201 1524 367 1598">getScreentip (getScreentip callback)</p>	<p data-bbox="456 1524 1390 1577">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 1587 1446 1640">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 data-bbox="456 1650 1003 1671">For example, consider the following XML fragment:</p> <pre data-bbox="509 1703 1224 1724" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getScreentip="GetButtonScreentip" /></pre>


Attributes	Description
	<p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 541 1247 562" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getShowImage="IsButtonImageVisible" /> </pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 926 1247 947" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getShowLabel="IsButtonLabelVisible" /> </pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1310 1198 1331" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getSupertip="GetButtonSupertip" /> </pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1694 1159 1715" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getVisible="IsButtonVisible" /> </pre> <p>In this example, the IsButtonVisible callback function is called when the application needs</p>

Attributes	Description
	<p>to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 506 1008 531"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 873 816 898"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1268 1300 1577"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>

Attributes	Description
image (custom image identifier)	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 407 1019 428" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" image="ForestPic" /> </pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 825 995 846" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" imageMso="Bold" /> </pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1234 1273 1304" style="background-color: #f0f0f0; padding: 5px;"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1692 1287 1761" style="background-color: #f0f0f0; padding: 5px;"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre>

Attributes	Description
	<p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 533 1289 604"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 995 1305 1066"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1528 1198 1558"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1070 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" label="Custom Button" /></pre> <p data-bbox="456 296 1349 380">This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 407 324 457">screentip (screentip)</p>	<p data-bbox="456 407 1446 516">Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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="456 527 1097 548">For example, consider a button with a screentip, as follows:</p> <div data-bbox="456 558 951 894" style="border: 1px solid #ccc; padding: 5px;">  </div> <p data-bbox="456 905 1000 926">This is specified using the following XML fragment:</p> <pre data-bbox="509 953 1208 999" style="background-color: #f0f0f0; padding: 5px;"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p data-bbox="456 1058 1349 1100">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1127 422 1178">showImage (show image)</p>	<p data-bbox="456 1127 1357 1236">Specifies whether this control displays an icon. The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1226 1235 1247">For example, consider a button that does not display an icon, as follows:</p> <div data-bbox="456 1257 699 1425" style="border: 1px solid #ccc; padding: 5px;">  </div> <p data-bbox="456 1436 1000 1457">This is specified using the following XML fragment:</p> <pre data-bbox="509 1484 980 1530" style="background-color: #f0f0f0; padding: 5px;"><button id="button" showImage="false" label="Button with no icon" /></pre> <p data-bbox="456 1625 1406 1646">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 1673 406 1724">showLabel (show label)</p>	<p data-bbox="456 1673 1053 1694">Specifies whether this control SHOULD display its label.</p> <p data-bbox="456 1705 1430 1759">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p data-bbox="456 1770 1005 1791">For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1159 264"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p data-bbox="457 317 1406 369">This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p data-bbox="457 380 1406 405">The possible values for this attribute are defined by the XML schema boolean datatype.</p>
<p data-bbox="201 428 422 453">supertip (supertip)</p>	<p data-bbox="457 428 1114 453">Specifies a string to be shown as the supertip of the control.</p> <p data-bbox="457 464 1398 516">The supertip and getSupertip attributes are mutually exclusive. If neither attribute is specified no supertip for this control SHOULD be shown.</p> <p data-bbox="457 527 1092 552">For example, consider a control with a supertip, as follows:</p>  <p data-bbox="457 947 1003 972">This is specified using the following XML fragment:</p> <pre data-bbox="509 999 1208 1066"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p data-bbox="457 1119 1349 1171">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1194 305 1220">tag (tag)</p>	<p data-bbox="457 1194 1390 1272">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 data-bbox="457 1283 1398 1308">If this attribute is omitted, the control's tag value SHOULD default to an empty string.</p> <p data-bbox="457 1318 1008 1344">For example, consider the following XML fragment:</p> <pre data-bbox="509 1371 1105 1417"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p data-bbox="457 1470 1281 1522">This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p data-bbox="457 1533 1349 1585">The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p data-bbox="201 1604 375 1656">visible (control visibility)</p>	<p data-bbox="457 1604 911 1629">Specifies the visibility state of the control.</p> <p data-bbox="457 1640 1365 1692">The getVisible and visible attributes are mutually exclusive. If these attributes are omitted, the control SHOULD default to being visible.</p> <p data-bbox="457 1703 1008 1728">For example, consider the following XML fragment:</p> <pre data-bbox="509 1755 1008 1780"><tab idMso="TabHome" visible="false" /></pre>

Attributes	Description
	In this example, the built-in tab with an identifier of "TabHome" is hidden. 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:

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

For example, consider a split button control, as follows:



Figure 23: A split button control

This is 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>
```

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the child elements of this element.

Child Elements	Section
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

The following table summarizes the attributes of this element.


Attributes	Description
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getImage (getImage callback)	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p>


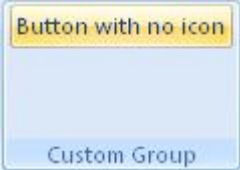
Attributes	Description
	<pre data-bbox="509 218 1146 241" style="text-align: center;"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p data-bbox="456 296 1390 348">In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p data-bbox="456 359 1385 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p data-bbox="456 432 1435 516">Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p data-bbox="456 527 1005 550">For example, consider the following XML fragment:</p> <pre data-bbox="509 579 1122 602" style="text-align: center;"><button id="button" getLabel="GetButtonLabel" /></pre> <p data-bbox="456 657 1451 709">In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p data-bbox="456 720 1385 772">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p data-bbox="456 789 1395 842">Specifies the name of a callback function to be called to determine the screentip of this control.</p> <p data-bbox="456 852 1446 926">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 data-bbox="456 936 1005 959">For example, consider the following XML fragment:</p> <pre data-bbox="509 989 1224 1012" style="text-align: center;"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p data-bbox="456 1066 1427 1119">In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p data-bbox="456 1129 1385 1182">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p data-bbox="456 1199 1435 1251">Specifies the name of a callback function to be called to determine whether the application SHOULD display the icon of this control.</p> <p data-bbox="456 1262 1390 1314">This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 1325 1357 1377">The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p data-bbox="456 1388 1005 1411">For example, consider the following XML fragment:</p> <pre data-bbox="509 1440 1248 1463" style="text-align: center;"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p data-bbox="456 1518 1333 1570">In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p data-bbox="456 1581 1385 1633">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel (getShowLabel callback)	<p data-bbox="456 1640 1435 1692">Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</p> <p data-bbox="456 1703 1390 1755">This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p data-bbox="456 1766 1430 1818">The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p>



Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 249 1248 275"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSize (getSize callback)</p>	<p>Specifies the name of a callback function to be called to determine the size of this control. The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 611 1094 636"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 995 1198 1020"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1377 1159 1402"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>id (control identifier)</p>	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1008 241" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 296 1390 380">This specifies a custom button control with an identifier of "MyButton". The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 407 1422 558">Specifies the identifier of a built-in control. The contents of this attribute are application-defined. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 585 816 609" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 663 1433 747">This creates a clone of the control with an identifier of "Bold". The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 770 1433 951">Specifies a qualified identifier for a control. The idQ attribute can be used to reference controls or containers created by other Custom UI documents. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 978 1299 1287" style="background-color: #f0f0f0; padding: 5px;"><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 data-bbox="456 1341 1430 1505">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab. The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 1530 1442 1703">Specifies the relationship identifier 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. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1730 1019 1753" style="background-color: #f0f0f0; padding: 5px;"><button id="button" image="ForestPic" /></pre>

Attributes	Description
	<p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 541 997 569"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 951 1273 1024"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1409 1289 1482"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1287 323"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 716 1300 785"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1251 1198 1274"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1549 1070 1572"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD display the label of the control as the screentip or display</p>

Attributes	Description
	<p>no screentip at all. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 659 1211 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>showImage (show image)</p>	<p>Specifies whether this control displays an icon. This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large". The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon. For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1255 980 1306"><button id="button" showImage="false" label="Button with no icon" /></pre> <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 size or getSize attributes specify that the control is "large". The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label. For example, consider the following XML fragment:</p> <pre data-bbox="509 1646 1159 1696"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p>

Attributes	Description
<p>size (control size)</p>	<p>The possible values for this attribute are defined by the XML schema boolean datatype.</p> <p>Specifies the size of the control.</p> <p>The size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 611 1045 636"><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
<p>supertip (supertip)</p>	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1331 1208 1402"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1703 1105 1749"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre>

Attributes	Description
	This example is a button with a tag value of "123456", which is passed to the ButtonClicked callback function. The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.
visible (control visibility)	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. For example, consider the following XML fragment: <pre><tab idMso="TabHome" visible="false" /></pre> In this example, the built-in tab with an identifier of "TabHome" is hidden. 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:

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

2.2.39 tab (Tab)

This element specifies a ribbon tab control.

For example, consider the following XML fragment:

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

This XML fragment specifies a custom tab with the **label** "My Custom Tab".

The following table summarizes the elements that are parents of this element.

Parent Elements
tabs (section 2.2.40); tabSet (section 2.2.41)

The following table summarizes the child elements of this element.


Child Elements	Section
group (Group)	2.2.23

The following table summarizes the attributes of this element.

Attributes	Description
getKeytip	Specifies the name of a callback function to be called to determine the suggested KeyTip of

Attributes	Description
(getKeytip callback)	<p>this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1146 369"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 701 1122 726"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to be called to determine the visibility state of this control. The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible. For example, consider the following XML fragment:</p> <pre data-bbox="509 1087 1159 1113"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified. For example, consider the following XML fragment:</p> <pre data-bbox="509 1497 1008 1522"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an identifier of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p>Specifies the identifier of a built-in control. The contents of this attribute are application-defined. The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 816 275" style="margin-left: 40px;"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 646 1300 953" style="margin-left: 40px;"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1394 1276 1465" style="margin-left: 40px;"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 254 1284 323"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>insertBeforeMso (identifier of built-in control to insert before)</p>	<p>Specifies the identifier of a built-in control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 716 1284 785"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertBeforeQ (qualified identifier of control to insert before)</p>	<p>Specifies the qualified identifier of a control that this control SHOULD be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1167 1284 1236"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>keytip (keytip)</p>	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1709 1195 1732"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as</p>

Attributes	Description
	specified in section 2.3.7 .
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 394 1068 422"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 751 1105 800"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1136 1003 1163"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
ribbon (section 2.2.33)

The following table summarizes the child elements of this element.

Child Elements	Section
tab (Tab)	2.2.39

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

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

For example, consider the following XML fragment:

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

This XML fragment is used to add a new custom tab to the tab set with an identifier of "TabSetPictureTools".

The following table summarizes the elements that are parents of this element.

Parent Elements
contextualTabs (section 2.2.10)

The following table summarizes the child elements of this element.

Child Elements	Subclause
tab (Tab)	section 2.2.39

The following table summarizes the attributes of this element.

Attributes	Description
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 380 1159 401" style="text-align: center;"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
idMso (built-in control identifier)	<p>Specifies the identifier of a built-in control.</p> <p>The contents of this attribute are application-defined.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 716 818 737" style="text-align: center;"><control idMso="Bold" /></pre> <p>This is used to create a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1013 1062" style="text-align: center;"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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


2.2.42 toggleButton (Unsize Button)

This element specifies a toggle button control that, because of its location, cannot have its size changed. The **size** attribute is not present. It otherwise behaves identically to the regular **toggleButton** element, as specified in section [2.2.43](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
buttonGroup (section 2.2.5); menu (section 2.2.28); menu (section 2.2.26); menu (section 2.2.29); menu (section 2.2.27); officeMenu (section 2.2.31)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which SHOULD be displayed in detailed views.</p> <p>The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control.</p> <p>The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getEnabled (getEnabled callback)	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is</p>



Attributes	Description
	<p>specified, the control SHOULD default to being enabled. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1159 310"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 642 1122 667"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control. The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider the following XML fragment:</p> <pre data-bbox="509 1031 1146 1056"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getLabel (getLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 1122 1413"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getPressed (getPressed callback)</p>	<p>Specifies the name of a callback function to be called to determine the toggled state of this control. If this attribute is omitted, the control SHOULD default to the off state. For example, consider the following XML fragment:</p> <pre data-bbox="509 1745 1235 1770"><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre>


Attributes	Description
	<p>In this example, the IsButtonToggled callback function is called when the application needs to determine the toggle state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getScreentip (getScreentip callback)</p>	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 533 1224 554" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getScreentip="GetButtonScreentip" /> </pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to 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 is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 974 1248 995" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getShowImage="IsButtonImageVisible" /> </pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to 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 is "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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1415 1248 1436" style="background-color: #f0f0f0; padding: 5px;"> <button id="button" getShowLabel="IsButtonLabelVisible" /> </pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p>


Attributes	Description
	<pre data-bbox="509 218 1198 241" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p data-bbox="456 296 1414 348">In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p data-bbox="456 359 1382 411">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p data-bbox="456 432 1446 485">Specifies the name of a callback function to be called to determine the visibility state of this control.</p> <p data-bbox="456 495 1349 548">The getVisible and visible attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being visible.</p> <p data-bbox="456 558 1000 579">For example, consider the following XML fragment:</p> <pre data-bbox="509 602 1159 625" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 680 1446 732">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 743 1382 795">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 810 1406 894">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 905 1422 957">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 968 1000 989">For example, consider the following XML fragment:</p> <pre data-bbox="509 1012 1005 1035" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 1089 1216 1110">This specifies a custom button control with an identifier of "MyButton".</p> <p data-bbox="456 1121 1390 1173">The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 1199 919 1220">Specifies the identifier of a built-in control.</p> <p data-bbox="456 1230 1032 1251">The contents of this attribute are application-defined.</p> <p data-bbox="456 1262 1422 1314">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1325 1000 1346">For example, consider the following XML fragment:</p> <pre data-bbox="509 1369 818 1392" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 1457 1118 1478">This creates a clone of the control with an identifier of "Bold".</p> <p data-bbox="456 1488 1430 1541">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 1556 919 1577">Specifies a qualified identifier for a control.</p> <p data-bbox="456 1587 1430 1640">The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p data-bbox="456 1650 1422 1703">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1713 1000 1734">For example, consider the following XML fragment:</p> <pre data-bbox="509 1757 1300 1810" style="background-color: #f0f0f0; padding: 5px;"><customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"</pre>

Attributes	Description
	<pre data-bbox="509 201 1105 464"> 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 data-bbox="456 516 1430 617">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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p data-bbox="456 627 1425 680">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
image (custom image identifier)	<p data-bbox="456 705 1442 779">Specifies the relationship identifier 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 data-bbox="456 789 1438 842">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 852 1003 877">For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1019 926"> <button id="button" image="ForestPic" /> </pre> <p data-bbox="456 978 1438 1031">This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p data-bbox="456 1041 1442 1094">The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p data-bbox="456 1117 1446 1199">Specifies the identifier 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 data-bbox="456 1209 1438 1262">The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p data-bbox="456 1272 1003 1297">For example, consider the following XML fragment:</p> <pre data-bbox="509 1325 997 1346"> <button id="button" imageMso="Bold" /> </pre> <p data-bbox="456 1398 1393 1451">This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p data-bbox="456 1461 1438 1514">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p data-bbox="456 1537 1446 1589">Specifies the identifier of a built-in control that this control SHOULD be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1600 1438 1673">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 data-bbox="456 1684 1003 1709">For example, consider the following XML fragment:</p> <pre data-bbox="509 1736 1273 1778"> <tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </pre>

Attributes	Description
	<p data-bbox="509 201 586 222"></tab></p> <p data-bbox="456 279 1425 327">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 338 1430 386">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 411 407 516">insertAfterQ (qualified identifier of control to insert after)</p>	<p data-bbox="456 411 1419 464">Specifies the qualified identifier of a control that this control SHOULD be inserted after. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 474 1435 548">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 data-bbox="456 558 1003 579">For example, consider the following XML fragment:</p> <pre data-bbox="509 611 1284 680"> <tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 737 1425 785">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 795 1425 844">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 867 418 972">insertBeforeMso (identifier of built-in control to insert before)</p>	<p data-bbox="456 867 1419 919">Specifies the identifier of a built-in control that this control SHOULD be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 930 1435 1003">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 data-bbox="456 1014 1003 1035">For example, consider the following XML fragment:</p> <pre data-bbox="509 1066 1284 1136"> <tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 1192 1442 1241">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 1251 1430 1299">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p data-bbox="201 1323 407 1428">insertBeforeQ (qualified identifier of control to insert before)</p>	<p data-bbox="456 1323 1435 1375">Specifies the qualified identifier of a control that this control SHOULD be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 1386 1435 1459">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 data-bbox="456 1470 1003 1491">For example, consider the following XML fragment:</p> <pre data-bbox="509 1522 1284 1591"> <tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab> </pre> <p data-bbox="456 1648 1442 1696">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 1707 1425 1755">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p data-bbox="201 1778 370 1799">keytip (keytip)</p>	<p data-bbox="456 1778 1208 1799">Specifies a string to be used as the suggested KeyTip for this control.</p>

Attributes	Description
	<p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically. For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 443 1198 468"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
<p>label (label)</p>	<p>Specifies a string that SHOULD be used as the label for this control. The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed. For example, consider the following XML fragment:</p> <pre data-bbox="509 743 1068 768"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button". The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>onAction (onAction callback)</p>	<p>Specifies the name of a callback function to be called when this control is invoked by the user. For example, consider the following XML fragment:</p> <pre data-bbox="509 1041 1300 1066"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>In this example, the button calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>screentip (screentip)</p>	<p>Specifies a string to be shown as the screentip for this control. The screentip and getScreentip 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. For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1776 1208 1801"><button id="button" imageMso="HappyFace" label="Button"</pre>

Attributes	Description
	<pre>size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <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 size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 638 1208 709"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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. For example, consider the following XML fragment:</p> <pre data-bbox="509 1010 1105 1058"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1394 1008 1415"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

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


2.2.43 toggleButton (Toggle Button)

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

For example, consider a toggle button control, as follows:



Figure 24: A toggle button control


This is specified with the following XML fragment:

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

The following table summarizes the elements that are parents of this element.

Parent Elements
box (section 2.2.1); group (section 2.2.23)

The following table summarizes the attributes of this element.

Attributes	Description
description (description)	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider a button with a detailed description, as follows:</p>  <p>This is 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>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
enabled (enabled state)	<p>Specifies the enabled state of the control. The getEnabled and enabled 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 that would otherwise be disabled by the application.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Disabled Button" enabled="false" /></pre>


Attributes	Description
	<p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used.</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 to be called to determine the detailed description of this control.</p> <p>The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 478 1273 506"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control.</p> <p>The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 863 1159 890"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getImage (getImage callback)</p>	<p>Specifies the name of a callback function to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1220 1122 1247"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getKeytip (getKeytip callback)</p>	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1604 1146 1631"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>


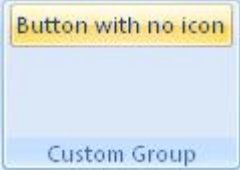
Attributes	Description
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1122 380"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getPressed (getPressed callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 709 1235 737"><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p>In this example, the IsButtonToggled callback function is called when the application needs to determine the toggle state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1119 1224 1146"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowImage (getShowImage callback)	<p>Specifies the name of a callback function to be called to determine whether the application displays the icon of this control.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1566 1248 1593"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getShowLabel	<p>Specifies the name of a callback function to be called to determine whether the application</p>



Attributes	Description
(getShowLabel callback)	<p>displays the label of this control.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre data-bbox="509 407 1247 428" style="text-align: center;"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSize (getSize callback)	<p>Specifies the name of a callback function to be called to determine the size of this control.</p> <p>The getSize and size attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 764 1094 785" style="text-align: center;"><button id="button" getSize="GetButtonSize" /></pre> <p>In this example, the GetButtonSize callback function is called when the application needs to determine the size of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getSupertip (getSupertip callback)	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1146 1198 1167" style="text-align: center;"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getVisible (getVisible callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1533 1159 1554" style="text-align: center;"><button id="button" getVisible="IsButtonVisible" /></pre> <p>In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p>Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify</p>

Attributes	Description
	<p>which control corresponds to the function call.</p> <p>The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 344 1008 369"><button id="MyButton" label="Button" /></pre> <p>This specifies a custom button control with an id of "MyButton".</p> <p>The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
<p>idMso (built-in control identifier)</p>	<p>Specifies the identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 711 816 737"><control idMso="Bold" /></pre> <p>This creates a clone of the control with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>idQ (qualified control identifier)</p>	<p>Specifies a qualified identifier 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 MUST be specified.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1104 1300 1413"><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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier for an image to 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, no icon SHOULD be displayed.</p>

Attributes	Description
	<p>For example, consider the following XML fragment:</p> <pre data-bbox="509 247 1019 275"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon is the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST_Uri simple type, as specified in section 2.3.14.</p>
imageMso (built-in image identifier)	<p>Specifies the identifier of a built-in image to 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 667 997 695"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that uses the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterMso (identifier of built-in control to insert after)	<p>Specifies the identifier of a built-in control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1052 1273 1125"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertAfterQ (qualified identifier of control to insert after)	<p>Specifies the qualified identifier of a control that this control is to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1514 1289 1587"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in	<p>Specifies the identifier of a built-in control that this control is to be inserted before. If the</p>

Attributes	Description
control to insert before)	<p>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>For example, consider the following XML fragment:</p> <pre data-bbox="509 373 1287 443"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p>Specifies the qualified identifier of a control that this control is to be inserted before. 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 831 1300 900"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an id of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p>Specifies a string to be used as the suggested KeyTip for this control.</p> <p>The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider a button with KeyTip 'K', as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1371 1198 1394"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p>The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1671 1070 1694"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as</p>

Attributes	Description
	specified in section 2.3.11 .
onAction (onAction callback)	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>This specifies a button that calls the ButtonClicked callback function when it is invoked. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "large".</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><button id="button" showImage="false" label="Button with no icon" /></pre> <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 displays its label.</p> <p>This attribute SHOULD have no effect if the size or getSize attributes specify that the control is "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>For example, consider the following XML fragment:</p> <pre><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</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 size and getSize attributes are mutually exclusive. If neither attribute is specified, the control's size SHOULD default to the normal size.</p> <p>For example, consider a large button, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre><toggleButton idMso="Bold" size="large" /></pre> <p>The possible values for this attribute are defined by the ST_Size simple type, as specified in section 2.3.10.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is 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
	The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.
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. For example, consider the following XML fragment:</p> <pre data-bbox="509 449 1105 495"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 831 1008 856"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT ToggleButton">
  <xsd:complexContent>
    <xsd:extension base="CT ToggleButtonRegular">
      <xsd:attributeGroup ref="AG SizeAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd: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 in the same way as the regular **toggleButton** element, as specified in section [2.2.43](#).

The following table summarizes the elements that are parents of this element.

Parent Elements
splitButton (section 2.2.38); splitButton (section 2.2.36); splitButton (section 2.2.37)

The following table summarizes the attributes of this element.


Attributes	Description
<p>description (description)</p>	<p>Specifies a detailed description of the control, which is displayed in detailed views. The description and getDescription attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider a button with a detailed description, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 499 1263 569"><button id="button" label="Button" imageMso="HappyFace" description="This is a verbose description that describes the function of this control in detail." /></pre> <p>The possible values for this attribute are defined by the ST_LongString simple type, as specified in section 2.3.8.</p>
<p>enabled (enabled state)</p>	<p>Specifies the enabled state of the control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled. This attribute cannot be used to enable a built-in control that would otherwise be disabled by the application. For example, consider the following XML fragment:</p> <pre data-bbox="509 905 1300 926"><button id="button" label="Disabled Button" enabled="false" /></pre> <p>This specifies a new button that is disabled. A permanently disabled button is not very useful, thus the enabled attribute is not commonly used. 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 to be called to determine the detailed description of this control. The getDescription and description attributes are mutually exclusive. If neither attribute is specified, the control SHOULD NOT display any detailed text. For example, consider the following XML fragment:</p> <pre data-bbox="509 1262 1273 1283"><button id="button" getDescription="GetButtonDescription" /></pre> <p>In this example, the GetButtonDescription callback function is called when the application needs to determine the detailed description of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getEnabled (getEnabled callback)</p>	<p>Specifies the name of a callback function to be called to determine the enabled state of this control. The getEnabled and enabled attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to being enabled. For example, consider the following XML fragment:</p> <pre data-bbox="509 1646 1159 1667"><button id="button" getEnabled="IsButtonEnabled" /></pre> <p>In this example, the IsButtonEnabled callback function is called when the application needs to determine the enabled state of the button. The possible values for this attribute are defined by the ST_Delegate simple type, as</p>


Attributes	Description
	specified in section 2.3.2.
getImage (getImage callback)	<p>Specifies the name of a callback function to be called to determine the icon of this control. The getImage, image, and imageMso attributes are mutually exclusive. If none of these attributes are specified, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 394 1122 422"><button id="button" getImage="GetButtonImage" /></pre> <p>In this example, the GetButtonImage callback function is called when the application needs to determine the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getKeytip (getKeytip callback)	<p>Specifies the name of a callback function to be called to determine the suggested KeyTip of this control.</p> <p>The getKeytip and keytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 779 1146 806"><button id="button" getKeytip="GetButtonKeytip" /></pre> <p>In this example, the GetButtonKeytip callback function is called when the application needs to determine the KeyTip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getLabel (getLabel callback)	<p>Specifies the name of a callback function to be called to determine the label of this control. The getLabel and label attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1136 1122 1163"><button id="button" getLabel="GetButtonLabel" /></pre> <p>In this example, the GetButtonLabel callback function is called when the application needs to determine the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getPressed (getPressed callback)	<p>Specifies the name of a callback function to 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1493 1235 1520"><toggleButton id="toggle" getPressed="IsButtonToggled" /></pre> <p>In this example, the IsButtonToggled callback function is called when the application needs to determine the toggle state of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
getScreentip (getScreentip callback)	<p>Specifies the name of a callback function to 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</p>



Attributes	Description
	<p>no screentip at all.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1224 310"><button id="button" getScreentip="GetButtonScreentip" /></pre> <p>In this example, the GetButtonScreentip callback function is called when the application needs to determine the screentip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowImage (getShowImage callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application displays the icon of this control.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 669 1252 695"><button id="button" getShowImage="IsButtonImageVisible" /></pre> <p>In this example, the IsButtonImageVisible callback function is called when the application needs to determine whether to display the icon of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getShowLabel (getShowLabel callback)</p>	<p>Specifies the name of a callback function to be called to determine whether the application SHOULD display the label of this control.</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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1054 1252 1079"><button id="button" getShowLabel="IsButtonLabelVisible" /></pre> <p>In this example, the IsButtonLabelVisible callback function is called when the application needs to determine whether to display the label of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getSupertip (getSupertip callback)</p>	<p>Specifies the name of a callback function to 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> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 1438 1198 1463"><button id="button" getSupertip="GetButtonSupertip" /></pre> <p>In this example, the GetButtonSupertip callback function is called when the application needs to determine the supertip of the button.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
<p>getVisible (getVisible callback)</p>	<p>Specifies the name of a callback function to be called to determine the visibility state of this control. This attribute is prohibited and the visibility is controlled by the split button.</p> <p>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1159 239" style="background-color: #f0f0f0; padding: 5px;"><button id="button" getVisible="IsButtonVisible" /></pre> <p data-bbox="456 296 1448 348">In this example, the IsButtonVisible callback function is called when the application needs to determine the visibility of the button.</p> <p data-bbox="456 357 1383 409">The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
id (control identifier)	<p data-bbox="456 434 1406 510">Specifies the identifier for a custom control. All custom controls MUST have unique identifiers. The identifier of a control SHOULD be passed to callback functions to identify which control corresponds to the function call.</p> <p data-bbox="456 518 1422 571">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 579 1005 604">For example, consider the following XML fragment:</p> <pre data-bbox="509 630 1005 651" style="background-color: #f0f0f0; padding: 5px;"><button id="MyButton" label="Button" /></pre> <p data-bbox="456 707 1216 732">This specifies a custom button control with an identifier of "MyButton".</p> <p data-bbox="456 741 1390 793">The possible values for this attribute are defined by the ST_UniqueID simple type, as specified in section 2.3.13.</p>
idMso (built-in control identifier)	<p data-bbox="456 819 919 844">Specifies the identifier of a built-in control.</p> <p data-bbox="456 852 1029 877">The contents of this attribute are application-defined.</p> <p data-bbox="456 886 1422 938">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 947 1005 972">For example, consider the following XML fragment:</p> <pre data-bbox="509 997 815 1018" style="background-color: #f0f0f0; padding: 5px;"><control idMso="Bold" /></pre> <p data-bbox="456 1075 1118 1100">This creates a clone of the control with an identifier of "Bold".</p> <p data-bbox="456 1108 1433 1161">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
idQ (qualified control identifier)	<p data-bbox="456 1182 919 1207">Specifies a qualified identifier for a control.</p> <p data-bbox="456 1215 1433 1268">The idQ attribute can be used to reference controls or containers created by other Custom UI documents.</p> <p data-bbox="456 1276 1422 1329">The id, idQ, and idMso attributes are mutually exclusive. At least one of these attributes MUST be specified.</p> <p data-bbox="456 1337 1005 1362">For example, consider the following XML fragment:</p> <pre data-bbox="509 1388 1299 1696" style="background-color: #f0f0f0; padding: 5px;"><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 data-bbox="456 1753 1433 1806">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 identifier of "OtherTab". If</p>

Attributes	Description
	<p>that tab cannot be found, it is created. A new group belonging to this file is added to the tab.</p> <p>The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
<p>image (custom image identifier)</p>	<p>Specifies the relationship identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 533 1019 558"><button id="button" image="ForestPic" /></pre> <p>This specifies a custom button whose icon SHOULD be the embedded image file referenced by the relationship identifier of "ForestPic".</p> <p>The possible values for this attribute are defined by the ST Uri simple type, as specified in section 2.3.14.</p>
<p>imageMso (built-in image identifier)</p>	<p>Specifies the identifier 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, no icon SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 953 993 978"><button id="button" imageMso="Bold" /></pre> <p>This specifies a custom button that SHOULD use the built-in image with an identifier of "Bold".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterMso (identifier of built-in control to insert after)</p>	<p>Specifies the identifier of a built-in control that 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>For example, consider the following XML fragment:</p> <pre data-bbox="509 1360 1273 1432"><tab id="MyTab" insertAfterMso="TabHome" label="Custom Tab"> ... </tab></pre> <p>In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the built-in tab with an identifier of "TabHome".</p> <p>The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
<p>insertAfterQ (qualified identifier of control to insert after)</p>	<p>Specifies the qualified identifier of a control that 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>For example, consider the following XML fragment:</p>

Attributes	Description
	<pre data-bbox="509 218 1289 289"><tab id="MyTab" insertAfterQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 344 1425 394">In this example, a new custom tab with an identifier of "MyTab" is to be inserted after the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 403 1425 453">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
insertBeforeMso (identifier of built-in control to insert before)	<p data-bbox="456 480 1425 531">Specifies the identifier of a built-in control that this control SHOULD be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 539 1438 617">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 data-bbox="456 625 1005 651">For example, consider the following XML fragment:</p> <pre data-bbox="509 678 1289 749"><tab id="MyTab" insertBeforeMso="TabHome" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 804 1443 854">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the built-in tab with an identifier of "TabHome".</p> <p data-bbox="456 863 1435 913">The possible values for this attribute are defined by the ST_ID simple type, as specified in section 2.3.5.</p>
insertBeforeQ (qualified identifier of control to insert before)	<p data-bbox="456 938 1438 989">Specifies the qualified identifier of a control that this control SHOULD be inserted before. If the value of this attribute is not understood, it SHOULD be ignored.</p> <p data-bbox="456 997 1438 1075">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 data-bbox="456 1083 1005 1108">For example, consider the following XML fragment:</p> <pre data-bbox="509 1136 1289 1207"><tab id="MyTab" insertBeforeQ="x:OtherTab" label="Custom Tab"> ... </tab></pre> <p data-bbox="456 1262 1443 1312">In this example, a new custom tab with an identifier of "MyTab" is to be inserted before the custom tab with a qualified identifier of "x:OtherTab".</p> <p data-bbox="456 1320 1425 1371">The possible values for this attribute are defined by the ST_QID simple type, as specified in section 2.3.9.</p>
keytip (keytip)	<p data-bbox="456 1396 1211 1421">Specifies a string to be used as the suggested KeyTip for this control.</p> <p data-bbox="456 1430 1346 1480">The keytip and getKeytip attributes are mutually exclusive. If neither attribute is specified, the application SHOULD generate a KeyTip for the control automatically.</p> <p data-bbox="456 1488 1086 1514">For example, consider a button with KeyTip 'K', as follows:</p>  <p data-bbox="456 1621 1002 1646">This is specified using the following XML fragment:</p> <pre data-bbox="509 1673 1198 1701"><button id="button" imageMso="HappyFace" keytip="K" /></pre> <p data-bbox="456 1755 1354 1806">The possible values for this attribute are defined by the ST_Keytip simple type, as specified in section 2.3.7.</p>

Attributes	Description
label (label)	<p>Specifies a string to be used as the label for this control.</p> <p>The label and getLabel attributes are mutually exclusive. If neither attribute is specified, no label SHOULD be displayed.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 352 1068 378"><button id="button" label="Custom Button" /></pre> <p>This specifies a custom button with a label of "Custom Button".</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
onAction (onAction callback)	<p>Specifies the name of a callback function to be called when this control is invoked by the user.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 653 1300 678"><button id="button" label="Button" onAction="ButtonClicked" /></pre> <p>In this example, the button calls the ButtonClicked callback function when it is invoked.</p> <p>The possible values for this attribute are defined by the ST_Delegate simple type, as specified in section 2.3.2.</p>
screentip (screentip)	<p>Specifies a string to be shown as the screentip for this control.</p> <p>The screentip and getScreentip 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>For example, consider a button with a screentip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1388 1208 1436"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="This is the screentip" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
showImage (show image)	<p>Specifies whether this control displays an icon.</p> <p>The showImage and getShowImage attributes are mutually exclusive. If neither attribute is specified, the control SHOULD display its icon.</p> <p>For example, consider a button that does not display an icon, as follows:</p>

Attributes	Description
	 <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 432 980 478"><button id="button" showImage="false" label="Button with no icon" /></pre> <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>The showLabel and getShowLabel attributes are mutually exclusive. If neither attribute is specified, the control SHOULD default to showing its label.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 726 1159 772"><button id="button" label="Label" showLabel="false" imageMso="HappyFace" /></pre> <p>This specifies a button that has a label, but does not show it. Even though the label is hidden, it is provided to accessibility tools.</p> <p>The possible values for this attribute are defined by the XML schema boolean datatype.</p>
supertip (supertip)	<p>Specifies a string to 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 for this control SHOULD be shown.</p> <p>For example, consider a control with a supertip, as follows:</p>  <p>This is specified using the following XML fragment:</p> <pre data-bbox="509 1503 1208 1575"><button id="button" imageMso="HappyFace" label="Button" size="large" screentip="Screentip" supertip="This is the supertip string" /></pre> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 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>

Attributes	Description
	<p>If this attribute is omitted, the control's tag value SHOULD default to an empty string. For example, consider the following XML fragment:</p> <pre data-bbox="509 285 1105 331"><button id="button" label="Button" tag="123456" onAction="ButtonClicked" /></pre> <p>This specifies a button with a tag value of "123456", which is passed to the ButtonClicked callback function.</p> <p>The possible values for this attribute are defined by the ST_String simple type, as specified in section 2.3.11.</p>
<p>visible (control visibility)</p>	<p>Specifies the visibility state of the control. This attribute is prohibited and the visibility is controlled by the split button.</p> <p>For example, consider the following XML fragment:</p> <pre data-bbox="509 636 1008 657"><tab idMso="TabHome" visible="false" /></pre> <p>In this example, the built-in tab with an identifier of "TabHome" is hidden.</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:

```
<xsd:complexType name="CT_VisibleToggleButton">
  <xsd:complexContent>
    <xsd:restriction base="CT_ToggleButtonRegular">
      <xsd:attribute name="visible" use="prohibited"/>
      <xsd:attribute name="getVisible" use="prohibited"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd: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 are laid out horizontally.
vertical (Vertical)	Specifies that the child controls are laid out vertically.

Referenced By

box@boxStyle (section [2.2.1](#))

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

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

2.3.2 ST_Delegate (Callback Function Name)

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

Examples of this simple type are **macro** scripts and **add-in** callback functions.

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 (section [2.2.1](#)); button@getDescription (section 2.2.2); button@getDescription (section [2.2.3](#)); button@getDescription (section [2.2.4](#)); button@getEnabled (section 2.2.2); button@getEnabled (section 2.2.3); button@getEnabled (section 2.2.4); button@getImage (section 2.2.2); button@getImage (section 2.2.3); button@getImage (section 2.2.4); button@getKeytip (section 2.2.2); button@getKeytip (section 2.2.3); button@getKeytip (section 2.2.4); button@getLabel (section 2.2.2); button@getLabel (section 2.2.3); button@getLabel (section 2.2.4); button@getScreentip (section 2.2.2); button@getScreentip (section 2.2.3); button@getScreentip (section 2.2.4); button@getShowImage (section 2.2.2); button@getShowImage (section 2.2.3); button@getShowImage (section 2.2.4); button@getShowLabel (section 2.2.2); button@getShowLabel (section 2.2.3); button@getShowLabel (section 2.2.4); button@getSize (section 2.2.2); button@getSupertip (section 2.2.2); button@getSupertip (section 2.2.3); button@getSupertip (section 2.2.4); button@getVisible (section 2.2.2); button@getVisible (section 2.2.3); button@getVisible (section 2.2.4); button@onAction (section 2.2.2); button@onAction (section 2.2.3); button@onAction (section 2.2.4); buttonGroup@getVisible (section [2.2.5](#)); checkBox@getDescription (section [2.2.6](#)); checkBox@getEnabled (section 2.2.6); checkBox@getImage (section 2.2.6); checkBox@getKeytip (section 2.2.6); checkBox@getLabel (section 2.2.6); checkBox@getPressed (section 2.2.6); checkBox@getScreentip (section 2.2.6); checkBox@getShowImage (section 2.2.6); checkBox@getShowLabel (section 2.2.6); checkBox@getSupertip (section 2.2.6); checkBox@getVisible (section 2.2.6); checkBox@onAction (section 2.2.6); comboBox@getEnabled (section [2.2.7](#)); comboBox@getImage (section 2.2.7); comboBox@getItemCount (section 2.2.7); comboBox@getItemID (section 2.2.7); comboBox@getItemImage (section 2.2.7); comboBox@getItemLabel (section 2.2.7); comboBox@getItemScreentip (section 2.2.7); comboBox@getItemSupertip (section 2.2.7); comboBox@getKeytip (section 2.2.7); comboBox@getLabel (section 2.2.7); comboBox@getScreentip (section 2.2.7); comboBox@getShowImage (section 2.2.7); comboBox@getShowLabel (section 2.2.7); comboBox@getSupertip (section 2.2.7); comboBox@getText (section 2.2.7); comboBox@getVisible (section 2.2.7); comboBox@onChange (section 2.2.7); command@getEnabled (section [2.2.8](#)); command@onAction (section 2.2.8); control@getDescription (section [2.2.12](#)); control@getDescription (section [2.2.13](#)); control@getEnabled (section 2.2.12); control@getEnabled (section 2.2.13); control@getEnabled (section [2.2.11](#)); control@getImage (section 2.2.12); control@getImage (section 2.2.13); control@getImage (section 2.2.11); control@getKeytip (section 2.2.12); control@getKeytip (section 2.2.13); control@getKeytip (section 2.2.11); control@getLabel (section 2.2.12); control@getLabel (section 2.2.13); control@getLabel (section 2.2.11); control@getScreentip (section 2.2.12); control@getScreentip (section 2.2.13); control@getScreentip (section 2.2.11); control@getShowImage (section 2.2.12); control@getShowImage (section 2.2.13); control@getShowImage (section 2.2.11);

Referenced By

control@showLabel (section 2.2.12); control@showLabel (section 2.2.13); control@showLabel (section 2.2.11); control@getSize (section 2.2.12); control@getSize (section 2.2.13); control@getSupertip (section 2.2.12); control@getSupertip (section 2.2.13); control@getSupertip (section 2.2.11); control@getVisible (section 2.2.12); control@getVisible (section 2.2.13); control@getVisible (section 2.2.11); control@onAction (section 2.2.12); customUI@loadImage (section [2.2.14](#)); customUI@onLoad (section 2.2.14); dropDown@getEnabled (section [2.2.17](#)); dropDown@getImage (section 2.2.17); dropDown@getItemCount (section 2.2.17); dropDown@getItemID (section 2.2.17); dropDown@getItemImage (section 2.2.17); dropDown@getItemLabel (section 2.2.17); dropDown@getItemScreentip (section 2.2.17); dropDown@getItemSupertip (section 2.2.17); dropDown@getKeytip (section 2.2.17); dropDown@getLabel (section 2.2.17); dropDown@getScreentip (section 2.2.17); dropDown@getSelectedItemID (section 2.2.17); dropDown@getSelectedItemIndex (section 2.2.17); dropDown@getShowImage (section 2.2.17); dropDown@getShowLabel (section 2.2.17); dropDown@getSupertip (section 2.2.17); dropDown@getVisible (section 2.2.17); dropDown@onAction (section 2.2.17); dynamicMenu@getContent (section [2.2.19](#)); dynamicMenu@getContent (section [2.2.18](#)); dynamicMenu@getDescription (section 2.2.19); dynamicMenu@getDescription (section 2.2.18); dynamicMenu@getEnabled (section 2.2.19); dynamicMenu@getEnabled (section 2.2.18); dynamicMenu@getImage (section 2.2.19); dynamicMenu@getImage (section 2.2.18); dynamicMenu@getKeytip (section 2.2.19); dynamicMenu@getKeytip (section 2.2.18); dynamicMenu@getLabel (section 2.2.19); dynamicMenu@getLabel (section 2.2.18); dynamicMenu@getScreentip (section 2.2.19); dynamicMenu@getScreentip (section 2.2.18); dynamicMenu@getShowImage (section 2.2.19); dynamicMenu@getShowImage (section 2.2.18); dynamicMenu@getShowLabel (section 2.2.19); dynamicMenu@getShowLabel (section 2.2.18); dynamicMenu@getSize (section 2.2.19); dynamicMenu@getSupertip (section 2.2.19); dynamicMenu@getSupertip (section 2.2.18); dynamicMenu@getVisible (section 2.2.19); dynamicMenu@getVisible (section 2.2.18); editBox@getEnabled (section [2.2.20](#)); editBox@getImage (section 2.2.20); editBox@getKeytip (section 2.2.20); editBox@getLabel (section 2.2.20); editBox@getScreentip (section 2.2.20); editBox@getShowImage (section 2.2.20); editBox@getShowLabel (section 2.2.20); editBox@getSupertip (section 2.2.20); editBox@getText (section 2.2.20); editBox@getVisible (section 2.2.20); editBox@onChange (section 2.2.20); gallery@getDescription (section [2.2.21](#)); gallery@getDescription (section [2.2.22](#)); gallery@getEnabled (section 2.2.21); gallery@getEnabled (section 2.2.22); gallery@getImage (section 2.2.21); gallery@getImage (section 2.2.22); gallery@getItemCount (section 2.2.21); gallery@getItemCount (section 2.2.22); gallery@getItemHeight (section 2.2.21); gallery@getItemHeight (section 2.2.22); gallery@getItemID (section 2.2.21); gallery@getItemID (section 2.2.22); gallery@getItemImage (section 2.2.21); gallery@getItemImage (section 2.2.22); gallery@getItemLabel (section 2.2.21); gallery@getItemLabel (section 2.2.22); gallery@getItemScreentip (section 2.2.21); gallery@getItemScreentip (section 2.2.22); gallery@getItemSupertip (section 2.2.21); gallery@getItemSupertip (section 2.2.22); gallery@getItemWidth (section 2.2.21); gallery@getItemWidth (section 2.2.22); gallery@getKeytip (section 2.2.21); gallery@getKeytip (section 2.2.22); gallery@getLabel (section 2.2.21); gallery@getLabel (section 2.2.22); gallery@getScreentip (section 2.2.21); gallery@getScreentip (section 2.2.22); gallery@getSelectedItemID (section 2.2.21); gallery@getSelectedItemID (section 2.2.22); gallery@getSelectedItemIndex (section 2.2.21); gallery@getSelectedItemIndex (section 2.2.22); gallery@getShowImage (section 2.2.21); gallery@getShowImage (section 2.2.22); gallery@getShowLabel (section 2.2.21); gallery@getShowLabel (section 2.2.22); gallery@getSize (section 2.2.21); gallery@getSupertip (section 2.2.21); gallery@getSupertip (section 2.2.22); gallery@getVisible (section 2.2.21); gallery@getVisible (section 2.2.22); gallery@onAction (section 2.2.21); gallery@onAction (section 2.2.22); group@getImage (section [2.2.23](#)); group@getKeytip (section 2.2.23); group@getLabel (section 2.2.23); group@getScreentip (section 2.2.23); group@getSupertip (section 2.2.23); group@getVisible (section 2.2.23); labelControl@getEnabled (section [2.2.25](#)); labelControl@getImage (section 2.2.25); labelControl@getKeytip (section 2.2.25); labelControl@getLabel (section 2.2.25); labelControl@getScreentip (section 2.2.25); labelControl@getShowImage (section 2.2.25); labelControl@getShowLabel (section 2.2.25); labelControl@getSupertip (section 2.2.25); labelControl@getVisible (section 2.2.25); menu@getDescription (section [2.2.28](#)); menu@getDescription (section [2.2.26](#)); menu@getEnabled (section 2.2.28); menu@getEnabled (section 2.2.26); menu@getEnabled (section [2.2.27](#)); menu@getImage (section 2.2.28); menu@getImage (section 2.2.26); menu@getImage (section 2.2.27); menu@getKeytip (section 2.2.28); menu@getKeytip (section 2.2.26); menu@getKeytip (section 2.2.27); menu@getLabel (section 2.2.28); menu@getLabel (section 2.2.26); menu@getLabel (section 2.2.27); menu@getScreentip (section 2.2.28); menu@getScreentip (section 2.2.26); menu@getScreentip (section 2.2.27); menu@getShowImage (section 2.2.28); menu@getShowImage (section 2.2.26); menu@getShowImage (section 2.2.27); menu@getShowLabel (section 2.2.28); menu@getShowLabel (section 2.2.26); menu@getShowLabel (section 2.2.27); menu@getSize (section 2.2.28); menu@getSupertip (section 2.2.28); menu@getSupertip (section 2.2.26); menu@getSupertip (section 2.2.27); menu@getTitle (section [2.2.29](#)); menu@getTitle (section 2.2.27); menu@getVisible (section 2.2.28); menu@getVisible (section 2.2.26); menu@getVisible (section 2.2.27); menuSeparator@getTitle (section [2.2.30](#)); separator@getVisible (section [2.2.34](#)); splitButton@getEnabled (section [2.2.38](#)); splitButton@getEnabled (section [2.2.36](#)); splitButton@getEnabled (section [2.2.37](#)); splitButton@getImage (section 2.2.38); splitButton@getImage (section 2.2.36); splitButton@getImage (section 2.2.37); splitButton@getKeytip (section 2.2.38); splitButton@getKeytip (section 2.2.36); splitButton@getKeytip (section 2.2.37); splitButton@getLabel (section 2.2.38); splitButton@getLabel (section 2.2.36);

Referenced By

splitButton@getLabel (section 2.2.37); splitButton@getScreentip (section 2.2.38); splitButton@getScreentip (section 2.2.36); splitButton@getScreentip (section 2.2.37); splitButton@getShowImage (section 2.2.38); splitButton@getShowImage (section 2.2.36); splitButton@getShowImage (section 2.2.37); splitButton@getShowLabel (section 2.2.38); splitButton@getShowLabel (section 2.2.36); splitButton@getShowLabel (section 2.2.37); splitButton@getSize (section 2.2.38); splitButton@getSupertip (section 2.2.38); splitButton@getSupertip (section 2.2.36); splitButton@getSupertip (section 2.2.37); splitButton@setVisible (section 2.2.38); splitButton@setVisible (section 2.2.36); splitButton@setVisible (section 2.2.37); tab@getKeytip (section 2.2.39); tab@getLabel (section 2.2.39); tab@setVisible (section 2.2.39); tabSet@setVisible (section 2.2.41); toggleButton@getDescription (section 2.2.43); toggleButton@getDescription (section 2.2.42); toggleButton@getDescription (section 2.2.44); toggleButton@setEnabled (section 2.2.43); toggleButton@setEnabled (section 2.2.42); toggleButton@setEnabled (section 2.2.44); toggleButton@getImage (section 2.2.43); toggleButton@getImage (section 2.2.42); toggleButton@getImage (section 2.2.44); toggleButton@getKeytip (section 2.2.43); toggleButton@getKeytip (section 2.2.42); toggleButton@getKeytip (section 2.2.44); toggleButton@getLabel (section 2.2.43); toggleButton@getLabel (section 2.2.42); toggleButton@getLabel (section 2.2.44); toggleButton@getPressed (section 2.2.43); toggleButton@getPressed (section 2.2.42); toggleButton@getPressed (section 2.2.44); toggleButton@getScreentip (section 2.2.42); toggleButton@getScreentip (section 2.2.44); toggleButton@getShowImage (section 2.2.43); toggleButton@getShowImage (section 2.2.42); toggleButton@getShowImage (section 2.2.44); toggleButton@getShowLabel (section 2.2.43); toggleButton@getShowLabel (section 2.2.42); toggleButton@getShowLabel (section 2.2.44); toggleButton@getSize (section 2.2.43); toggleButton@getSupertip (section 2.2.43); toggleButton@getSupertip (section 2.2.42); toggleButton@getSupertip (section 2.2.44); toggleButton@setVisible (section 2.2.43); toggleButton@setVisible (section 2.2.42); toggleButton@setVisible (section 2.2.44); toggleButton@onAction (section 2.2.43); toggleButton@onAction (section 2.2.42); toggleButton@onAction (section 2.2.44)

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

```
<xsd:simpleType name="ST_Delegate">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd: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 (section 2.2.21); gallery@itemHeight (section 2.2.22); gallery@itemWidth (section 2.2.21); gallery@itemWidth (section 2.2.22)

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

```

<xsd:simpleType name="ST_GalleryItemWidthHeight">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="4096"/>
  </xsd:restriction>
</xsd: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 (section [2.2.21](#)); gallery@columns (section [2.2.22](#)); gallery@rows (section [2.2.21](#)); gallery@rows (section [2.2.22](#))

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

```

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

```

2.3.5 ST_ID (Control ID)

Specifies the identifier of a built-in control. The format of this **string** is defined by per application by the Custom UI Control identifier Tables, as specified in section [3](#).

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

Referenced By

box@insertAfterMso (section [2.2.1](#)); box@insertBeforeMso (section [2.2.1](#)); button@idMso (section [2.2.2](#)); button@idMso (section [2.2.3](#)); button@idMso (section [2.2.4](#)); button@imageMso (section [2.2.2](#)); button@imageMso (section [2.2.3](#)); button@imageMso (section [2.2.4](#)); button@insertAfterMso (section [2.2.2](#)); button@insertAfterMso (section [2.2.3](#)); button@insertAfterMso (section [2.2.4](#)); button@insertBeforeMso (section [2.2.2](#)); button@insertBeforeMso (section [2.2.3](#)); button@insertBeforeMso (section [2.2.4](#)); buttonGroup@insertAfterMso (section [2.2.5](#)); buttonGroup@insertBeforeMso (section [2.2.5](#)); checkBox@idMso (section [2.2.6](#)); checkBox@imageMso (section [2.2.6](#)); checkBox@insertAfterMso (section [2.2.6](#)); checkBox@insertBeforeMso (section [2.2.6](#)); comboBox@idMso (section [2.2.7](#)); comboBox@imageMso (section [2.2.7](#)); comboBox@insertAfterMso (section [2.2.7](#)); comboBox@insertBeforeMso (section [2.2.7](#)); command@idMso (section [2.2.8](#)); control@id (section [2.2.13](#)); control@idMso (section [2.2.12](#)); control@idMso (section [2.2.13](#)); control@idMso (section [2.2.11](#)); control@imageMso (section [2.2.12](#)); control@imageMso (section [2.2.13](#)); control@imageMso (section [2.2.11](#)); control@insertAfterMso (section [2.2.12](#)); control@insertAfterMso (section [2.2.13](#)); control@insertBeforeMso (section [2.2.12](#)); control@insertBeforeMso (section [2.2.13](#));

Referenced By

2.2.13); control@insertAfterMso (section 2.2.11); control@insertBeforeMso (section 2.2.12); control@insertBeforeMso (section 2.2.13); control@insertBeforeMso (section 2.2.11); dropDown@idMso (section 2.2.17); dropDown@imageMso (section 2.2.17); dropDown@insertAfterMso (section 2.2.17); dropDown@insertBeforeMso (section 2.2.17); dynamicMenu@idMso (section 2.2.19); dynamicMenu@idMso (section 2.2.18); dynamicMenu@imageMso (section 2.2.19); dynamicMenu@imageMso (section 2.2.18); dynamicMenu@insertAfterMso (section 2.2.19); dynamicMenu@insertAfterMso (section 2.2.18); dynamicMenu@insertBeforeMso (section 2.2.19); dynamicMenu@insertBeforeMso (section 2.2.18); editBox@idMso (section 2.2.20); editBox@imageMso (section 2.2.20); editBox@insertAfterMso (section 2.2.20); editBox@insertBeforeMso (section 2.2.20); gallery@idMso (section 2.2.21); gallery@idMso (section 2.2.22); gallery@imageMso (section 2.2.21); gallery@imageMso (section 2.2.22); gallery@insertAfterMso (section 2.2.21); gallery@insertAfterMso (section 2.2.22); gallery@insertBeforeMso (section 2.2.21); gallery@insertBeforeMso (section 2.2.22); group@idMso (section 2.2.23); group@imageMso (section 2.2.23); group@insertAfterMso (section 2.2.23); group@insertBeforeMso (section 2.2.23); item@imageMso (section 2.2.24); labelControl@idMso (section 2.2.25); labelControl@imageMso (section 2.2.25); labelControl@insertAfterMso (section 2.2.25); labelControl@insertBeforeMso (section 2.2.25); menu@idMso (section 2.2.28); menu@idMso (section 2.2.26); menu@idMso (section 2.2.27); menu@imageMso (section 2.2.28); menu@imageMso (section 2.2.26); menu@imageMso (section 2.2.27); menu@insertAfterMso (section 2.2.28); menu@insertAfterMso (section 2.2.26); menu@insertAfterMso (section 2.2.27); menu@insertBeforeMso (section 2.2.28); menu@insertBeforeMso (section 2.2.26); menu@insertBeforeMso (section 2.2.27); menuSeparator@insertAfterMso (section 2.2.30); menuSeparator@insertBeforeMso (section 2.2.30); separator@insertAfterMso (section 2.2.34); separator@insertBeforeMso (section 2.2.34); splitButton@idMso (section 2.2.38); splitButton@idMso (section 2.2.36); splitButton@idMso (section 2.2.37); splitButton@imageMso (section 2.2.38); splitButton@imageMso (section 2.2.36); splitButton@imageMso (section 2.2.37); splitButton@insertAfterMso (section 2.2.38); splitButton@insertAfterMso (section 2.2.36); splitButton@insertAfterMso (section 2.2.37); splitButton@insertBeforeMso (section 2.2.38); splitButton@insertBeforeMso (section 2.2.36); splitButton@insertBeforeMso (section 2.2.37); tab@idMso (section 2.2.39); tab@insertAfterMso (section 2.2.39); tab@insertBeforeMso (section 2.2.39); tabSet@idMso (section 2.2.41); toggleButton@idMso (section 2.2.43); toggleButton@idMso (section 2.2.42); toggleButton@idMso (section 2.2.44); toggleButton@imageMso (section 2.2.43); toggleButton@imageMso (section 2.2.42); toggleButton@imageMso (section 2.2.44); toggleButton@insertAfterMso (section 2.2.43); toggleButton@insertAfterMso (section 2.2.42); toggleButton@insertAfterMso (section 2.2.44); toggleButton@insertBeforeMso (section 2.2.43); toggleButton@insertBeforeMso (section 2.2.42); toggleButton@insertBeforeMso (section 2.2.44)

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

```
<xsd:simpleType name="ST_ID">
  <xsd:restriction base="xsd:NCName">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd: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 have large sizes.
normal (Normal)	Specifies that the child controls have normal sizes.

Referenced By

menu@itemSize (section [2.2.28](#)); menu@itemSize (section [2.2.26](#)); menu@itemSize (section [2.2.29](#)); menu@itemSize (section [2.2.27](#))

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

```
<xsd:simpleType name="ST_ItemSize">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="normal"/>
    <xsd:enumeration value="large"/>
  </xsd:restriction>
</xsd: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 character.
- This simple type's contents have a maximum length of 3 characters.

Referenced By

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

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

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

2.3.8 ST_LongString (Long String)

Specifies a string that 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 character.
- This simple type's contents have a maximum length of 4096 characters.

Referenced By

button@description (section [2.2.2](#)); button@description (section [2.2.3](#)); button@description (section [2.2.4](#)); checkBox@description (section [2.2.6](#)); control@description (section [2.2.12](#)); control@description (section [2.2.13](#)); dynamicMenu@description (section [2.2.19](#)); dynamicMenu@description (section [2.2.18](#)); gallery@description (section [2.2.21](#)); gallery@description (section [2.2.22](#)); menu@description (section [2.2.28](#)); menu@description (section [2.2.26](#)); toggleButton@description (section [2.2.43](#)); toggleButton@description (section [2.2.42](#)); toggleButton@description (section [2.2.44](#))

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

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

2.3.9 ST_QID (Qualified Control ID)

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

If the namespace is equal to the Custom UI namespace, the qualified identifier references the application's built-in control set.

For 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** 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" />
```

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

For 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 identifier of "OtherTab". If that tab cannot be found, it is created. A new group belonging to this file is added to the **tab**.

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

Referenced By

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

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

```

<xsd:simpleType name="ST_QID">
  <xsd:restriction base="xsd:QName">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd: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 (section 2.2.2); control@size (section 2.2.12); control@size (section 2.2.13); dynamicMenu@size (section 2.2.19); gallery@size (section 2.2.21); menu@size (section 2.2.28); splitButton@size (section 2.2.38); toggleButton@size (section 2.2.43)

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

```

<xsd:simpleType name="ST_Size">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="normal"/>
    <xsd:enumeration value="large"/>
  </xsd:restriction>
</xsd: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 character.
- This simple type's contents have a maximum length of 1024 characters.

Referenced By
button@label (section 2.2.2); button@label (section 2.2.3); button@label (section 2.2.4); button@screenTip (section 2.2.2); button@screenTip (section 2.2.3); button@screenTip (section 2.2.4); button@supertip (section 2.2.2); button@supertip (section 2.2.3); button@supertip (section 2.2.4); button@tag (section 2.2.2); button@tag (section 2.2.3); button@tag (section 2.2.4); checkBox@label (section 2.2.6); checkBox@screenTip (section 2.2.6); checkBox@supertip (section 2.2.6); checkBox@tag (section 2.2.6); comboBox@label (section 2.2.7); comboBox@screenTip (section 2.2.7); comboBox@sizeString (section 2.2.7); comboBox@supertip (section 2.2.7); comboBox@tag (section 2.2.7); control@label (section 2.2.12); control@label (section 2.2.13); control@label

Referenced By

(section [2.2.11](#)); control@screentip (section 2.2.12); control@screentip (section 2.2.13); control@screentip (section 2.2.11); control@supertip (section 2.2.12); control@supertip (section 2.2.13); control@supertip (section 2.2.11); control@tag (section 2.2.12); control@tag (section 2.2.11); dropDown@label (section [2.2.17](#)); dropDown@screentip (section 2.2.17); dropDown@sizeString (section 2.2.17); dropDown@supertip (section 2.2.17); dropDown@tag (section 2.2.17); dynamicMenu@label (section [2.2.19](#)); dynamicMenu@label (section [2.2.18](#)); dynamicMenu@screentip (section 2.2.19); dynamicMenu@screentip (section 2.2.18); dynamicMenu@supertip (section 2.2.19); dynamicMenu@supertip (section 2.2.18); dynamicMenu@tag (section 2.2.19); dynamicMenu@tag (section 2.2.18); editBox@label (section [2.2.20](#)); editBox@screentip (section 2.2.20); editBox@sizeString (section 2.2.20); editBox@supertip (section 2.2.20); editBox@tag (section 2.2.20); gallery@label (section [2.2.21](#)); gallery@label (section [2.2.22](#)); gallery@screentip (section 2.2.21); gallery@screentip (section 2.2.22); gallery@sizeString (section 2.2.21); gallery@sizeString (section 2.2.22); gallery@supertip (section 2.2.21); gallery@supertip (section 2.2.22); gallery@tag (section 2.2.21); gallery@tag (section 2.2.22); group@label (section [2.2.23](#)); group@screentip (section 2.2.23); group@supertip (section 2.2.23); group@tag (section 2.2.23); item@label (section [2.2.24](#)); item@screentip (section 2.2.24); item@supertip (section 2.2.24); labelControl@label (section [2.2.25](#)); labelControl@screentip (section 2.2.25); labelControl@supertip (section 2.2.25); labelControl@tag (section 2.2.25); menu@label (section [2.2.28](#)); menu@label (section [2.2.26](#)); menu@label (section [2.2.27](#)); menu@screentip (section 2.2.28); menu@screentip (section 2.2.26); menu@screentip (section 2.2.27); menu@supertip (section 2.2.28); menu@supertip (section 2.2.26); menu@supertip (section 2.2.27); menu@tag (section 2.2.28); menu@tag (section 2.2.26); menu@tag (section 2.2.27); menu@title (section [2.2.29](#)); menu@title (section 2.2.27); menuSeparator@title (section [2.2.30](#)); splitButton@label (section [2.2.38](#)); splitButton@label (section [2.2.36](#)); splitButton@label (section [2.2.37](#)); splitButton@screentip (section 2.2.38); splitButton@screentip (section 2.2.36); splitButton@screentip (section 2.2.37); splitButton@supertip (section 2.2.38); splitButton@supertip (section 2.2.36); splitButton@supertip (section 2.2.37); splitButton@tag (section 2.2.38); splitButton@tag (section 2.2.36); splitButton@tag (section 2.2.37); tab@label (section [2.2.39](#)); tab@tag (section 2.2.39); toggleButton@label (section [2.2.43](#)); toggleButton@label (section [2.2.42](#)); toggleButton@label (section [2.2.44](#)); toggleButton@screentip (section 2.2.43); toggleButton@screentip (section 2.2.42); toggleButton@screentip (section 2.2.44); toggleButton@supertip (section 2.2.43); toggleButton@supertip (section 2.2.42); toggleButton@supertip (section 2.2.44); toggleButton@tag (section 2.2.43); toggleButton@tag (section 2.2.42); toggleButton@tag (section 2.2.44)

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

```
<xsd:simpleType name="ST_String">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd: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 (section [2.2.7](#)); editBox@maxLength (section [2.2.20](#))

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

```
<xsd:simpleType name="ST_StringLength">
```

```

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

```

2.3.13 ST_UniqueID (Custom Control ID)

Specifies a custom control identifier.

This simple type's contents are a restriction of the XML schema **identifier** datatype.

This simple type also specifies the following restrictions:

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

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

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

```

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

```

2.3.14 ST_Uri (Image Relationship ID)

Specifies the relationship identifier of a part that is the target of a relationship from the containing Custom UI document.

The target part is an image part type, as specified in [\[ECMA-376\]](#) Part 1 section 15.2.13.

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 (section 2.2.2); button@image (section 2.2.3); button@image (section 2.2.4); checkBox@image (section 2.2.6); comboBox@image (section 2.2.7); control@image (section 2.2.12); control@image (section 2.2.13); control@image (section 2.2.11); dropDown@image (section 2.2.17); dynamicMenu@image (section 2.2.19); dynamicMenu@image (section 2.2.18); editBox@image (section 2.2.20); gallery@image (section 2.2.21);

Referenced By

gallery@image (section [2.2.22](#)); group@image (section [2.2.23](#)); item@image (section [2.2.24](#)); labelControl@image (section [2.2.25](#)); menu@image (section [2.2.28](#)); menu@image (section [2.2.26](#)); menu@image (section [2.2.27](#)); splitButton@image (section [2.2.38](#)); splitButton@image (section [2.2.36](#)); splitButton@image (section [2.2.37](#)); toggleButton@image (section [2.2.43](#)); toggleButton@image (section [2.2.42](#)); toggleButton@image (section [2.2.44](#))

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

```
<xsd:simpleType name="ST_Uri">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd: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
FontSizeDecreaseWord	button	Shrink Font
FileClose	button	Close

idMso	Control Type	Label
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...
BorderInside	toggleButton	Inside Borders
BorderOutside	toggleButton	Outside Borders
BorderNone	toggleButton	No Border
MailMergeGoToFirstRecord	button	First

idMso	Control Type	Label
MailMergeGoToPreviousRecord	button	Previous
MailMergeGoToNextRecord	button	Next
MailMergeGotToLastRecord	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
CalloutOptions	button	Callout Options
DataFormAddRecord	button	Add
DataFormDeleteRecord	button	Delete
FieldsUpdate	button	Update
DatabaseInsert	button	Insert Database
GridSettings	button	Grid Settings...

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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...
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...

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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...
LineStyleDialog	button	More Lines...
ArrowsMore	button	More Arrows...
WordArtVerticalText	toggleButton	Vertical Text
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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:

idMso	Control Type	Label
MessageHeaderToggle	button	Message Header
BorderInsideHorizontal	toggleButton	Inside Horizontal Border
BorderInsideVertical	toggleButton	Inside Vertical Border
BorderDiagonalDown	toggleButton	Diagonal Down Border
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

idMso	Control Type	Label
WordArtEditTextClassic	button	Edit Text...
TableInsertCellsDialog	button	Insert Cells...
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
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

idMso	Control Type	Label
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...
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...

idMso	Control Type	Label
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
NextMisspelling	button	Next Misspelling
HyphenationManual	button	Manual
BulletsAndNumberingClassic	button	Tools Bullets Numbers
CompareAndCombine	button	Compare & Combine
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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 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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
SortDescendingWord	button	Sort Descending
FrameInsertHorizontal	button	Horizontal Frame
FieldCodesToggle	button	Toggle Field Codes
GoToFootnote	button	Go to Footnote
GoToEndnote	button	Go to Endnote
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

idMso	Control Type	Label
BordersShadingDialogWord	button	Borders and Shading...
TextBoxWordClassic	button	Text Box
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...
FileNewWebPage	button	New Web Page
FileNewBlankDocument	button	New Blank Document

idMso	Control Type	Label
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
WebControlTextBox	button	Textbox
WebControlTextArea	button	Text Area
WebControlSubmit	button	Submit
WebControlSubmitWithImage	button	Submit with Image

idMso	Control Type	Label
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
StylesPane	button	Styles...
DeleteStyle	button	Delete Style
StylesRenameStyle	button	Rename Style
SelectNumber	button	Select Number
NumberingRestart	button	Restart Numbering
DrawingCanvasInsert	button	New Drawing Canvas

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
GroupComments	group	Comments
GroupChanges	group	Changes
GroupCompare	group	Compare
GroupPictureSize	group	Size
GroupTableAlignment	group	Alignment
GroupTextBoxText	group	Text
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

idMso	Control Type	Label
NumberingGalleryWord	gallery	Numbering
GroupControls	group	Controls
GroupZoom	group	Zoom
ArrowStyleGallery	gallery	Arrows
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

idMso	Control Type	Label
ChartPrimaryVerticalAxis	gallery	Primary Vertical Axis
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
BulletDefineNew	button	Define New Bullet...

idMso	Control Type	Label
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 Office Online...
TabAddIns	tab	Add-Ins
ReviewShowMarkupMenu	menu	Show Markup
ObjectEditShapeMenu	menu	Edit Shape

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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 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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
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...
QuickStylesResetDocumentStyles	button	Reset Document Quick Styles
QuickStylesResetFromTemplate	button	Reset to Quick Styles from

idMso	Control Type	Label
		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...
TextEffectsBevelMore3DOptionsDialog	button	3-D Options...
TextEffects3DRotationGallery	gallery	3-D Rotation
TextEffects3DRotationOptionsDialog	button	3-D Rotation Options...
TextEffectGlowGallery	gallery	Glow
TextGlowColorPicker	gallery	More Glow Colors

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
Drawing1ColorPickerLineStyle	gallery	Picture Border
Drawing1GalleryTextures	gallery	Texture
InsertBuildingBlocksEquationsGallery	gallery	Equation
Drawing1GalleryBrightness	gallery	Brightness
Drawing1GalleryContrast	gallery	Contrast
GroupDiagramArrangeClassic	group	Arrange
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
PercentSign	button	Percent Sign
DollarSign	button	Dollar Sign
FunctionWizard	button	Insert Function...
ConstrainNumeric	button	Constrain Numeric
LightShading	button	Light Shading
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
FreezePanels	button	Freeze Panels
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

idMso	Control Type	Label
FileUpdate	button	Update File
ReadOnly	button	Toggle Read Only
AutoFilterClassic	button	AutoFilter
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

idMso	Control Type	Label
ClipArtInsert	toggleButton	Clip Art...
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
FormatCellsDialog	button	Format Cells...

idMso	Control Type	Label
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
SheetUnhide	button	Unhide Sheet...
SheetProtect	button	Protect Sheet...
ReviewProtectWorkbook	toggleButton	Protect Workbook...

idMso	Control Type	Label
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
CalculateSheet	button	Calculate Sheet
TextOrientationAngleCounterclockwise	toggleButton	Angle Counterclockwise
TextOrientationAngleClockwise	toggleButton	Angle Clockwise
WebGoBack	button	Back
WebGoForward	button	Forward

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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:
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...

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
GroupChartStyles	group	Chart Styles
GroupChartAxes	group	Axes
GroupChartShapes	group	Insert
GroupNumber	group	Number
GroupAlignmentExcel	group	Alignment
GroupCells	group	Cells
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

idMso	Control Type	Label
GroupZoom	group	Zoom
ArrowStyleGallery	gallery	Arrows
OutlineDashesGallery	gallery	Dashes
OutlineWeightGallery	gallery	Weight
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

idMso	Control Type	Label
OrientationMenu	menu	Orientation
MergeCenterMenu	splitButton	Merge
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
HeaderFooterCurrentDate	button	Current Date
HeaderFooterCurrentTimeInsert	button	Current Time

idMso	Control Type	Label
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
ConditionalFormattingHighlightBetween	button	Between...
ConditionalFormattingClearSelectedCells	button	Clear Rules from Selected Cells
ConditionalFormattingClearSheet	button	Clear Rules from Entire Sheet
ConditionalFormattingClearTable	button	Clear Rules from This Table

idMso	Control Type	Label
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
GroupSmartArtQuickStyles	group	SmartArt Styles
GroupSmartArtCreateGraphic	group	Create Graphic
GroupSmartArtReset	group	Reset
GroupSmartArtSize	group	Size
ConditionalFormattingHighlightCellsMenu	menu	Highlight Cells Rules
ConditionalFormattingTopBottomMenu	menu	Top/Bottom Rules

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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...
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...

idMso	Control Type	Label
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
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

idMso	Control Type	Label
GlowColorPicker	gallery	More Glow Colors
RecolorColorPicker	gallery	More Variations
GlowColorMoreColorsDialog	button	More Colors...
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

idMso	Control Type	Label
PivotTableSubtotalsOnBottom	button	Show all Subtotals at Bottom of Group
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
TableExportMenu	menu	Export

idMso	Control Type	Label
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
RefreshAllMenu	splitButton	Refresh
HideAndUnhideMenu	menu	Hide & Unhide
DataValidationMenu	splitButton	Data Validation

idMso	Control Type	Label
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
PasteAsPictureMenu	menu	As Picture
GroupPivotActions	group	Actions
PivotTableSelectFlyout	menu	Select
ZoomCurrent100	button	100%
TextFillMoreTextures	button	More Textures...

idMso	Control Type	Label
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
Clear	button	Clear
Superscript	toggleButton	Superscript
Subscript	toggleButton	Subscript
FileClose	button	Close
FormatPainter	toggleButton	Format Painter
FilePrintPreview	toggleButton	Print Preview

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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...
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

idMso	Control Type	Label
ObjectFillMoreColorsDialog	button	More Fill Colors...
ObjectBorderOutlineColorMoreColorsDialog	button	More Outline Colors...
LineStylesDialog	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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
ViewDisplayInPureBlackAndWhite	toggleButton	Pure Black and White
PrintWhat	dropDown	Print What
DrawingCanvasExpand	button	Expand
SlidesPerPage1Slide	toggleButton	1 Slide
SlideMasterPreserveMaster	toggleButton	Preserve
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

idMso	Control Type	Label
FilePermissionDoNotDistribute	toggleButton	Restricted Access
FilePermissionView	button	View Permission
Thesaurus	button	Thesaurus...
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

idMso	Control Type	Label
ShapesInsertGallery	gallery	Shapes
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
GroupSlideShowSetup	group	Set Up

idMso	Control Type	Label
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
GroupSize	group	Size
TabSlideMaster	tab	Slide Master
GroupTableStylesPowerPoint	group	Table Styles

idMso	Control Type	Label
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
ChartErrorBars	gallery	Error Bars
ChartLines	gallery	Lines
ChartUpDownBars	gallery	Up/Down Bars
ChartPlotArea	gallery	Plot Area
ChartWall	gallery	Chart Wall

idMso	Control Type	Label
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...
TabAddIns	tab	Add-Ins
ObjectEditShapeMenu	menu	Edit Shape
SlideMasterContentPlaceholderInsert	button	Content
SlideMasterTextPlaceholderInsert	button	Text
SlideMasterChartPlaceholderInsert	button	Chart
SlideMasterTablePlaceholderInsert	button	Table
SlideMasterDiagramPlaceholderInsert	button	SmartArt

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
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

idMso	Control Type	Label
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
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...

idMso	Control Type	Label
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
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

idMso	Control Type	Label
GlowColorPicker	gallery	More Glow Colors
RecolorColorPicker	gallery	More Variations
GlowColorMoreColorsDialog	button	More Colors...
PictureRecolorMoreColorsDialog	button	More Colors...
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

idMso	Control Type	Label
TextGlowColorPicker	gallery	More Glow Colors
TextGlowColorMoreColorsDialog	button	More Colors...
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
MenuPublish	menu	Publish
FileCompatibilityCheckerPowerPoint	button	Run Compatibility Checker

idMso	Control Type	Label
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
ZoomPageWidth
Zoom100
TableInsert

imageMso
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
HappyFace
UnderlineDouble
UnderlineWords
FontSizeIncreaseWord
FontSizeDecreaseWord

imageMso
_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
P
Q
R
S
T
U
V

imageMso
W
X
Y
Z
FileClose
TableAutoFormat
FormatPainter
FilePrintPreview
PickUpStyle
PasteApplyStyle
Bold
Italic
Underline
DarkShading
ParagraphMarks
AlignLeft
AlignRight
AlignCenter
AlignJustify
ContextHelp
HeaderFooterPageNumberInsert
Undo
Redo
Controlline
ControlRectangle
OutlinePromote
OutlineDemote
OutlineMoveUp
OutlineMoveDown
OutlineDemoteToBodyText
OutlineExpand
OutlineCollapse

imageMso
TextBoxInsert
FileFind
FindDialog
TableExcelSpreadsheetInsert
AutoFormat
BorderTop
BorderBottom
BorderLeft
BorderRight
BorderInside
BorderOutside
BorderNone
MailMergeGoToFirstRecord
MailMergeGoToPreviousRecord
MailMergeGoToNextRecord
MailMergeGotToLastRecord
MailMergeMergeToDocument
MailMergeMergeToPrinter
MailMergeAutoCheckForErrors
DataFormSource
MailMergeResultsPreview
ObjectsGroup
ObjectsUngroup
ObjectBringToFront
ObjectSendToBack
AboveText
BehindText
ObjectBringForward
ObjectSendBackward
Magnifier
PrintPreviewShrinkOnePage
MultiplePages

imageMso
ViewFullScreenView
ViewRulerPowerPoint
VoiceInsert
ObjectsSelect
TableFind
MacroRecord
MacroRecorderPause
MacroPlay
ObjectFlipHorizontal
ObjectFlipVertical
ObjectRotateRight90
ObjectRotateLeft90
ShapeFreeform
GroupDrawing
ObjectEditPoints
CalloutOptions
SortUp
SortDown
TableDesign
DataFormAddRecord
DataFormDeleteRecord
FieldsUpdate
DatabaseInsert
GridSettings
WordPicture
FormControlEditBox
FormControlCheckBox
FormControlComboBox
PropertySheet
FieldShading
ViewDraftView
Lock

imageMso
AutoSum
MasterDocumentShow
MasterDocumentCreateSubdocument
MasterDocumentUnlinkSubdocument
MasterDocumentInsertSubdocument
MasterDocumentSplitSubdocuments
MasterDocumentMergeSubdocuments
MasterDocumentLockSubdocument
HeaderOrFooterShow
HeaderFooterPreviousSection
HeaderFooterNextSection
AlignDialog
MailMergeDocument
MergeOptions
MailMergeHelper
PageSetupDialog
BodyTextHide
HeaderFooterLinkToPrevious
OutlineShowFirstLineOnly
OutlineShowTextFormatting
FontDialog
StylesDialogClassic
RoutingSlip
FootnoteInsert
MicrosoftExcel
MicrosoftAccess
Schedule
MicrosoftVisualFoxPro
MicrosoftPowerPoint
MicrosoftPublisher
MicrosoftProject
SadFace

imageMso
Pushpin
Camera
FormControlButton
Calculator
ViewPrintLayoutView
FieldCodes
DropCapOptionsDialog
Strikethrough
TextSmallCaps
CellsDelete
TableRowsDelete
TableColumnsDelete
CellsInsertDialog
TableRowsInsertWord
QueryInsertColumns
WindowsArrangeAll
MarginsAdjust
ViewGridlinesWord
SubdocumentOpen
WindowSplit
WindowNew
LegalBlackline
ReviewAcceptOrRejectChangeDialog
TextAllCaps
PictureDisassemble
SymbolInsert
ChangeCaseDialogClassic
FontSizeDecrease1Point
FontSizeIncrease1Point
Repaginate
ReplaceDialog
StartOfLine

imageMso
EndOfLine
PagePrevious
PageNext
TextBoxInsertVertical
StartOfDocument
EndOfDocument
Grammar
FileCloseOrCloseAll
TextToOrFromTable
TableRowsOrColumnsOrCellsInsert
TableRowsOrColumnsOrCellsDelete
RedoOrRepeat
ObjectsRegroup
_3DStyle
TipWizardHelp
AutoFormatChange
AddressBook
Reply
ReplyAll
Forward
MailMove
MailDelete
MessagePrevious
MessageNext
CheckNames
MailSelectNames
PrintAreaSetPrintArea
PasteFormatting
FillRight
FillDown
EqualSign
PlusSign

imageMso
MinusSign
MultiplicationSign
DivisionSign
ExponentiationSign
ParenthesisLeft
ParenthesisRight
ColonSign
CommaSign
PercentSign
DollarSign
FunctionWizard
AsianLayoutCharacterScaling
ConstrainNumeric
LightShading
AccountingFormat
PercentStyle
CommaStyle
DecimalsIncrease
DecimalsDecrease
MergeCenter
FontSizeIncrease
FontSizeDecrease
TextOrientationVertical
TextOrientationRotateUp
TextOrientationRotateDown
AlignDistributeHorizontallyClassic
ShapeScribble
ChartAreaChart
Chart3DBarChart
Chart3DColumnChart
Chart3DPieChart
ChartRadarChart

imageMso
OutlineSymbolsShowHide
TableSelectVisibleCells
SelectCurrentRegion
FreezePanels
ZoomIn
ZoomOut
FormControlRadioButton
FormControlScrollBar
FormControlListBox
TraceDependentRemoveArrows
TraceDependents
TracePrecedentsRemoveArrows
TraceRemoveAllArrows
FileUpdate
ReadOnly
AutoFilterClassic
Refresh
PivotTableFieldSettings
PivotTableShowPages
OutlineShowDetail
TraceError
OutlineHideDetail
AlignDistributeVerticallyClassic
FormControlGroupBox
FormControlSpinner
TabOrder
RunDialog
FormControlCombinationListEdit
FormControlCombinationDropDownEdit
FormControlLabel
Delete
Fish

imageMso
Coffee
Heart
Diamond
Spade
Club
ViewSheetGridlines
TracePrecedents
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

imageMso
QueryShowTable
QuerySelectQueryType
QueryCrosstab
QueryMakeTable
QueryUpdate
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

imageMso
FilterToggleFilter
FilterClearAllFilters
Head
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

imageMso
ParagraphSpacingIncrease
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
SelectAll

imageMso
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
ConvertTextToTable
TableFormulaDialog
TableSplitTable

imageMso
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
OutlineSettings
PrintPreviewClose
HeaderFooterClose
ZoomDialog
SortDialogClassic

imageMso
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
TextOrientationAngleCounterclockwise
TextOrientationAngleClockwise
HyperlinkOpenExcel
OpenStartPage
WebGoBack
WebGoForward
AddToFavorites

imageMso
BrowsePrevious
BrowseNext
BrowseSelector
SmartArtInsert
ShapeRerouteConnectors
ObjectNudgeUp
ObjectNudgeDown
ObjectNudgeLeft
ObjectNudgeRight
ShapeCurve
ShapeStraightConnector
ShapeElbowConnector
ObjectFillMoreColorsDialog
ObjectBorderOutlineColorMoreColorsDialog
LineStylesDialog
ArrowsMore
TextEffectAlignment
TextEffectTracking
WordArtVerticalText
WordArtEvenTextHeightClassic
ContrastMore
ContrastLess
BrightnessMore
BrightnessLess
ShadowNudgeUpClassic
ShadowNudgeDownClassic
ShadowNudgeLeftClassic
ShadowNudgeRightClassic
ObjectShadowColorMoreColorsDialog
HighImportance
LowImportance
AttachMenu

imageMso
InviteAttendees
AcceptInvitation
DeclineInvitation
TentativeAcceptInvitation
NewContact
NewTask
NewAppointment
TextAlignLeft
TextAlignCenter
ShapeRectangle
ShapeRoundedRectangle
ShapeIsoscelesTriangle
ShapeOval
ShapeSmileyFace
ShapeDonut
ShapeLeftBrace
ShapeRightBrace
ShapeArc
ShapeLightningBolt
ShapeHeart
ShapeRightArrow
ShapeLeftArrow
ShapeUpArrow
ShapeDownArrow
ShapeRoundedRectangularCallout
ShapeStar
ShapeSeal8
ShapeSeal16
ShapeSeal24
TextAlignRight
TextAlignLetterJustify
TextAlignWordJustify

imageMso
TextAlignStretchJustify
PictureReset
PictureRecolorAutomatic
PictureRecolorGrayscale
PictureRecolorBlackAndWhite
PictureRecolorWashout
TextWrappingSquare
TextWrappingTight
TextWrappingNoneClassic
TextWrappingEditWrapPoints
_3DEffectsOnOffClassic
_3DTiltDownClassic
_3DTiltUpClassic
_3DTiltLeftClassic
_3DTiltRightClassic
_3DExtrusionDirectionClassic
_3DLightingClassic
_3DSurfaceMaterialClassic
_3DExtrusionDepthNoneClassic
_3DExtrusionDepth36Classic
_3DExtrusionDepth72Classic
_3DExtrusionDepth144Classic
_3DExtrusionDepth288Classic
_3DExtrusionDepthInfinityClassic
_3DExtrusionPerspectiveClassic
_3DExtrusionParallelClassic
_3DLightingFlatClassic
_3DLightingNormalClassic
_3DLightingDimClassic
_3DSurfaceMatteClassic
_3DSurfacePlasticClassic
_3DSurfaceMetalClassic

imageMso
_3DSurfaceWireFrameClassic
ObjectEditText
SnapToShapes
TextWrappingMenuClassic
WindowsArrangeIcons
PictureFormatDialog
ViewVisualBasicCode
RemoveFromCalendar
MasterViewClose
CreateShortcutMenuFromMacro
DrawingNewClassic
HyperlinkInsert
HyperlinkEdit
ReviewNewComment
ReviewPreviousComment
ReviewNextComment
ReviewDeleteComment
ReviewShowOrHideComment
ReviewShowAllComments
PivotTableOptions
DesignMode
WordArtInsertDialogClassic
FormFieldProperties
PhoneticGuideEdit
FullScreenViewClassic
PhoneticGuideSettings
PhoneticGuideFieldShow
CircularReferences
MasterDocumentExpandOrCollapseSubdocuments
Chart3DConeChart
InternationalCurrency
ObjectsAlignCenterInFormHorizontally

imageMso
ObjectsAlignCenterInFormVertically
SizeToControlWidth
SizeToControlHeight
SizeToControlHeightAndWidth
HorizontalSpacingDecrease
HorizontalSpacingIncrease
ObjectsAlignDistributeHorizontallyRemove
VerticalSpacingDecrease
VerticalSpacingIncrease
ObjectsAlignDistributeVerticallyRemove
ObjectsArrangeBottom
ObjectsArrangeRight
CancelMeeting
Private
AcceptTask
SaveAndNew
CopyFolder
EmptyTrash
RecordInJournal
MarkAsUnread
CopyToFolder
MoveToFolder
ShapeFillColorPickerClassic
ControlLineColorPicker
VisualBasic
DoubleBottomBorder
BorderThickBottom
BorderTopAndBottom
BorderTopAndDoubleBottom
BorderTopAndThickBottom
BordersAll
BorderThickOutside

imageMso
SlideBackgroundFormatDialog
AutoSummarize
ViewDocumentMap
ReviewAcceptChange
ReviewRejectChange
TableDrawBorderPenStyle
Font
FontSize
ZoomClassic
CreateMap
MergeCellsAcross
FieldChooser
MessageHeaderToggle
MeetingRequest
NewNote
RecurrenceEdit
SendStatusReport
ImagerScan
QueryReturnGallery
SelectRecord
SelectAllRecords
TableTestValidationRules
RelationshipsClearLayout
SaveAsQuery
LoadFromQuery
DatasheetColumnRename
FileServerLinkTables
DatasheetColumnLookup
RecordsRefreshRecords
RelationshipsEditRelationships
RelationshipsHideTable
ReplicationRecoverDesignMaster

imageMso
ReplicationResolveConflicts
ReplicationCreateReplica
ReplicationSynchronizeNow
SetDatabasePassword
DatabaseUserLevelSecurityWizard
DatabaseUserAndGroupAccounts
ControlSpecialEffectEtched
ControlSpecialEffectShadowed
ControlSpecialEffectChiseled
WindowsCascade
PositionFitToWindow
BorderInsideHorizontal
BorderInsideVertical
BorderDiagonalDown
BorderDiagonalUp
MagicEightBall
TextDirectionLeftToRight
TextDirectionRightToLeft
ActiveXCheckBox
FindDialogExcel
ActiveXTextBox
ActiveXButton
ActiveXRadioButton
ActiveXListBox
ActiveXComboBox
ActiveXToggleButton
ActiveXSpinButton
ActiveXScrollBar
ActiveXLabel
ShowBcc
ShowFrom
DefinePrintStyles

imageMso
TagMarkComplete
NewContactFromSameCompany
ChooseForm
RecoverInviteToMeeting
FormPublish
SkipOccurrence
RightToLeftDocument
EditQuery
DataRangeProperties
RefreshAll
RefreshCancel
RefreshStatus
ClearAll
NewPostInThisFolder
ReplyToAttendeesWithMessage
SaveAndClose
AssignTask
Recurrence
NewMessageToContact
NewTaskForContact
NewMeetingWithContact
NewMessageToAttendees
SendUpdate
ReplyToAllAttendeesWithMessage
PostReplyToFolder
ViewAppointmentInCalendar
NewJournalEntry
NewMailMessage
CancelAcceptTask
CancelTaskAssignment
CancelDeclineTask
NewTaskRequest2

imageMso
RecurrenceEditSeries
DataValidation
DataValidationCircleInvalid
ReviewShareWorkbook
ReviewTrackChanges
ReviewHighlightChanges
DatabaseQueryNew
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

imageMso
WordArtEditTextClassic
FilterByResource
TableInsertCellsDialog
DeleteCells2
TableDeleteRows
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

imageMso
CompareAndCombine
PrintOptionsMenuWord
PageNumberFormat
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

imageMso
ShowAutoShapesAndDrawingBars
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
BlackAndWhiteBlackWithWhiteFill

imageMso
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
SlideShowCustom
DuplicateSelectedSlides
ActionInsert

imageMso
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
MicrosoftOnTheWeb06
MicrosoftOnTheWeb07
MicrosoftOnTheWeb08
MicrosoftOnTheWeb09
MicrosoftOnTheWeb10

imageMso
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
ObjectsAlignToGridOutlook
SortAscendingWord
SortDescendingWord
OutlineGroup
OutlineUngroup
IndentIncreaseExcel
IndentDecreaseExcel

imageMso
ControlAdvancedProperties
SaveAttachments
ViewVisualBasicCodeAccess
FrameInsertHorizontal
FormatPictureOrShapeDialogClassic
FileSendToPowerPoint
DeclineTask
AutoFormatNow
DataFormWord
ReviseContents
BulletListDefault
NumberListDefault
OutlineNumberDefault
FormatNumberDefault
FootnoteEndnoteDialog
TableInsertDialogWord
FormFieldClear
ObjectBringToFrontOfText
ObjectSendBehindText
RotateRight2
RotateLeft2
PageBreakInsertWord
BordersShadingDialogWord
TextBoxWordClassic
WordIndent
WordOutdent
IndentIncreaseWord
IndentDecreaseWord
SelectObjects
Callout
ListMacros
ReplaceWithAutoText

imageMso
FormatBackground
HangulHanjaConversion
HeaderSourceEdit
IndentIncrease
IndentDecrease
AsianLayoutFitText
AsianLayoutPhoneticGuide
AsianLayoutCombineCharacters
JapanesePostcardDialog
CharacterBorder
CharacterShading
MoviePlay
SlidesPerPage4Slides
SlidesPerPage9Slides
ViewWebLayoutView
PasteAlternative
PasteAsNestedTable
HyperlinkRemove
MacroSecurity
HorizontalLineInsert
WebPagePreview
TableSelectCell
TableDelete
TableInsertDialog
TableRowsInsertAboveWord
TableInsertRowsAbove
TableRowsInsertBelowWord
TableInsertRowsBelow
TableColumnsInsertLeft
TableInsertColumnsLeft
TableColumnsInsertRight
TableInsertColumnsRight

imageMso
TablePropertiesDialog
TableOptionsDialog
SendCopySendNow
SendCopySelectNames
SendCopyCheckNames
SendCopyOptions
SendCopyFlag
ConferenceMeetNow
SendCopySendToMailRecipient
TableOfContentsInFrame
ComAddInsDialog
FramePropertiesDialog
PivotChartInsertClassic
PivotFieldListShowHide
PictureBulletsInsert
FileNewWebPage
FileNewEmail
ToolboxAudio
ToolboxVideo
ToolboxMarquee
CreateStoredProcedure
FileNewBlankDocument
FileNewDialogClassic
ViewNormalViewPowerPoint
SynchronizeHtml
IgnoreHtmlChanges
FileSaveAsWebPage
HorizontalLineInsertClassic
GetExternalDataFromWeb
FrameCreateAbove
FrameCreateBelow
FrameCreateLeft

imageMso
FrameCreateRight
FrameDelete
RelationshipsReport
NewView
GetExternalDataFromText
ServerConnection
RecordsInsertSubdatasheet
AdpDiagramNewLabel
AdpDiagramAddRelatedTables
AdpDiagramShowRelationshipLabels
AdpDiagramViewPageBreaks
AdpDiagramRecalculatePageBreaks
AdpDiagramArrangeSelection
AdpDiagramArrangeTables
AdpDiagramNewTable
AdpDiagramColumnProperties
AdpDiagramColumnNames
AdpDiagramKeys
AdpDiagramNameOnly
AdpDiagramCustomView
AdpDiagramDeleteTable
AdpDiagramHideTable
AdpDiagramAutosizeSelectedTables
DiagramDeleteRelationship
WindowMoveSplit
EastAsianEditingMarks
FilePublishAsWebPage
TableAutoFitContents
TableAutoFitWindow
TableAutoFitFixedColumnWidth
TableCellAlignTopLeft
TableCellAlignTopCenter

imageMso
TableCellAlignTopRight
TableCellAlignMiddleLeft
TableCellAlignMiddleCenter
TableCellAlignMiddleRight
TableCellAlignBottomLeft
TableCellAlignBottomCenter
TableCellAlignBottomRight
ServerFilterApply
WebControlCheckBox
WebControlOptionButton
WebControlDropDownBox
WebControlListBox
WebControlTextBox
WebControlTextArea
WebControlSubmit
WebControlSubmitWithImage
WebControlReset
WebControlHidden
WebControlPassword
TableSetLeftToRight
TableSetRightToLeft
ChineseTranslationDialog
TableInsertMultidiagonalCell
AsianLayoutCharactersEnclose
ServerFilterByForm
EnvelopeChineseDialog
AdpVerifySqlSyntax
AdpDiagramTableModesMenu
CreateDiagram
RecordsExpandAllSubdatasheets
RecordsCollapseAllSubdatasheets
ObjectsMultiSelect

imageMso
DatabaseLinedTableManager
TranslateToTraditionalChinese
TextWrappingBehindText
TextWrappingInFrontOfText
WatermarkCustomDialog
FrameSaveCurrentAs
AdpOutputOperationsTableRemove
AdpOutputOperationsGroupBy
AdpViewDiagramPane
AdpViewGridPane
AdpViewSqlPane
AdpOutputOperationsAddToOutput
AdpOutputOperationsSortAscending
AdpOutputOperationsSortDescending
DatabaseAccessBackEnd
DatabasePartialReplica
TextDirectionContext
TranslateToSimplifiedChinese
ChineseTranslationMenu
FileBackUpSqlDatabase
ServerRestoreSqlDatabase
FileDropSqlDatabase
DatabaseSetLogonSecurity
DatabaseSqlServer
DatabaseSwitchboardManager
WebServerDiscussions
FontColorMoreColorsDialogPowerPoint
HyperlinksVerify
BlackAndWhite
CancelRequest
ViewMasterDocumentViewClassic
HyperlinkInsertPowerPoint

imageMso
AddOrRemoveAttendees
NewDistributionList
CalculateFull
SizeMenu
ScriptDebugger
RunThisForm
MessageFormatPlainText
MessageFormatHtml
MessageFormatRichText
ExportToVCardFile
MessageOptions
MapContactAddress
SignatureInsertMenu
DesignAFormOutlook
WatchWindow
FormulaEvaluate
LineSpacing
WordCountList
StylesModifyStyle
MacroRun
ServerProperties
SignatureShow
StylesPane
SpeechMicrophone
SubformInNewWindow
NumberingRestart
DrawingCanvasInsert
ViewsPivotTableView
ViewsPivotChartView
DiagramRadialInsertClassic
DiagramCycleInsertClassic
DiagramPyramidInsertClassic

imageMso
DiagramTargetInsertClassic
DiagramVennDiagramInsertClassic
DiagramChangeToRadialClassic
DiagramChangeToCycleClassic
DiagramChangeToTargetClassic
DiagramChangeToVennDiagramClassic
PasteByAppendingTable
OrganizationChartInsertAssistant
OrganizationChartInsertCoworker
OrganizationChartInsertSubordinate
AdpStoredProcedureEditSql
DrawingCanvasFit
DrawingCanvasResize
ColorGrayscaleMenu
ViewBackToColorView
ViewDisplayInGrayscale
ViewDisplayInHighContrast
ViewDisplayInPureBlackAndWhite
PivotAutoFilter
PivotSubtotal
PivotRefresh
PivotMoveToFieldArea
PivotMoveToColumnArea
PivotMoveToFilterArea
PivotMoveToDetailArea
PivotExpandField
PivotExportToExcel
PivotExpandIndicators
PivotDropAreas
SearchUI
MailMergeMergeToEMail
MailMergeMergeToFax

imageMso
MailMergeCreateList
DrawingCanvasExpand
TextWrappingInLineWithText
SlidesPerPage1Slide
ReviewSendForReview
WorkgroupAdmin
AdpDiagramAddTable
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

imageMso
PivotTableOlapPropertyFields
RevealFormatting
DiagramReverseClassic
DiagramAutoLayoutClassic
PasteTableFromExcel
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

imageMso
PageOrientationPortrait
GetExternalDataImportClassic
SlideReset
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

imageMso
DiagramFitToContentsClassic
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
ReviewDisplayForReview

imageMso
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
PersonaStatusOnline
PersonaStatusOffline
PersonaStatusAway

imageMso
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
TableExportTableToSharePointList
LookUp
FilePackageForCD
DesignXml
WindowSideBySide

imageMso
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
RmsImportanceUncheck
ViewThumbnails
Thesaurus
InkingStart
VoteMenu
RmsNavigationBar
RmsSendBizcard

imageMso
RmsSendBizcardDesign
FindText
AudioNoteDelete
ViewFullScreenReadingView
VisibilityVisible
VisibilityHidden
VisibilityInherit
InkDrawingAndWriting
ReviewInkCommentPen
ReviewInkCommentEraser
NewInternetFax
MeetingsWorkspace
InkEraser
InkStopInkingReadingView
StarUnratedEmpty
StarUnratedFull
StarRatedEmpty
StarRatedFull
StarRatedHalf
OutlookGlobe
OutlookGears
MsnLogo
GoLeftToRight
GoLtrHover
GoLtrFocus
GoLtrDown
GoRtl
GoRtlHover
GoRtlFocus
GoRtlDown
StopLeftToRight
StopLtrHover

imageMso
StopLtrFocus
StopLtrDown
StopRtl
StopRtlHover
StopRtlFocus
StopRtlDown
ForwardToMobile
JunkEmailAddToBlockedSendersList
JunkEmailMarkAsNotJunk
JunkEmailOptions
CategoryCollapse
DataSourceCatalogServerScript
PermissionRestrict
Risks
SetPertWeights
PanAndZoomWindow
FileBackupDatabase
OrganizationOutlook
ThemeBrowseForThemesPowerPoint
AccessNavigationOptions
WorkflowComplete
WorkflowPending
Connections
ViewGoBack
ViewGoForward
ListSetNumberingValue
SlideMasterInsertLayout
FileViewDigitalSignatures
SignaturesLoading
AppointmentAttachment
ViewsReportView
FileCloseDatabase

imageMso
OpenInfopathForm
FileWorkflowTasks
FileStartWorkflow
SharingRequestAllow
SharingRequestDeny
SharingOpenCalendarFolder
SignatureLineInsert
SlideThemesGallery
ApplicationOptionsDialog
BibliographyInsert
BibliographyStyle
CitationInsert
BibliographyManageSources
BibliographyAddNewSource
LabelInsert
BarcodeInsert
CongratulatoryEvent
CondolatoryEvent
ViewPageLayoutView
PivotClearAll
ChartStylesGallery
ChartLayoutGallery
ChartSaveTemplates
ChartAxisTitles
ChartAxes
ChartGridlines
ChartMoreElement
ChartFormatSelection
ChartElementSelector
PageMarginsGallery
LimeCatsAndDogsGallery
DropCapInsertGallery

imageMso
TrustCenter
ReadingViewAllowTyping
ReadingViewMarginSettingsMenu
CalendarInsert
SlideShowFromCurrent
BulletsGallery
NumberingGallery
LineSpacingGalleryPowerPoint
SlideNewGallery
SlideLayoutGallery
ShapesInsertGallery
ShapeChangeShapeGallery
ShapeFillTextureGallery
ShapeStylesGallery
PageOrientationGallery
FileServerTasksMenu
FileSendMenu
GroupFont
GroupParagraph
GroupStyles
GroupProofing
GroupInsertPages
GroupInsertIllustrations
GroupShapes
GroupWordArtText
GroupPageSetup
GroupParagraphLayout
GroupCitationsAndBibliography
GroupFootnotes
GroupTableOfContents
GroupMailMergeWriteInsertFields
GroupMailMergePreviewResults

imageMso
GroupMailMergeFinish
GroupChangesTracking
GroupComments
GroupChanges
GroupCompare
GroupPictureSize
GroupDrawBorders
GroupTableAlignment
GroupTableProperties
GroupTableTools
GroupSlides
GroupTextBoxText
GroupArrange
GroupInsertMediaClips
GroupSlideThemes
GroupThemes
GroupBackground
GroupPreview
GroupAnimations
GroupTransitionToThisSlide
GroupSlideShowStart
GroupSlideShowSetup
GroupShapeStyles
GroupFill
GroupLines
GroupEffects
GroupChartLayouts
GroupChartStyles
GroupChartAxes
GroupChartShapes
GroupNumber
GroupAlignmentExcel

imageMso
GroupCells
GroupSortFilter
GroupInsertTablesExcel
GroupOrganizationChartShapeInsert
GroupPageLayoutScaleToFit
GroupPageLayoutSheetOptions
GroupFunctionLibrary
GroupNamedCells
GroupFormulaAuditing
GroupGetExternalData
GroupConnections
GroupOutline
GroupDataTools
GroupChangesExcel
FollowUpReadMenu
StylesManageStyles
StylesStyleInspector
ObjectEffectPresetGallery
PictureEffectsPresetGallery
_3DRotationGallery
ControlLayoutTabular
ControlLayoutStacked
ControlLayoutRemove
ShapeFillColorPicker
OutlineColorPicker
ResultsPaneStartFindAndReplace
FileDocumentInspect
ImportSavedImports
ImportAccess
ImportExcel
ImportTextFile
ImportSharePointList

imageMso
ImportXmlFile
ImportOdbcDatabase
ImportHtmlDocument
ImportOutlook
ImportDBase
ImportParadox
ImportLotus
ExportSavedExports
ExportExcel
ExportSharePointList
ExportWord
ExportAccess
ExportTextFile
ExportXmlFile
ExportOdbcDatabase
ExportSnapshot
ExportHtmlDocument
ExportDBase
ExportParadox
ExportLotus
QuickStylesGallery
QuickStylesSets
ClearFormatting
FormRegionOpen
FormRegionSave
PanningHand
BulletsGalleryWord
NumberingGalleryWord
SharePointListsWorkOffline
SynchronizeData
SharePointListsDiscardAllChangesAndRefresh
FileManageMenu

imageMso
ViewsModeMenu
GroupSortAndFilter
SortSelectionMenu
FiltersMenu
RecordsMoreRecordsMenu
FieldsMenu
RecordsSubdatasheetMenu
GroupAutoFormatAccess
GroupTextFormatting
GroupDataTypeAndFormatting
AlignLeftToRightMenu
GroupCreateTables
GroupCreateForms
GroupCreateReports
GroupCreateOther
GroupSharepointLists
GroupCollectData
GroupImport
ImportMoreMenu
GroupExport
ExportMoreMenu
GroupDatabaseTools
DatabasePermissionsMenu
GroupMacro
GroupDatabaseSourceControl
GroupSourceControlShow
GroupSourceControlManage
GroupRelationships
GroupMacroRows
GroupQuerySetup
GroupQueryType
SqlSpecificMenu2

imageMso
GroupQueryTools
GroupControls
GroupPageLayoutAccess
GroupZoom
GroupToolsAccess
GroupRichText
GroupSizeAndPosition
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

imageMso
GroupPivotTableStyles
GroupPivotTableStyleOptions
WrapText
ClearMenu
ProofingToolsFlyoutAnchor
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

imageMso
ErrorCheckingMenu
TraceRemoveArrowsMenu
SortFilterMenu
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
ControllLineTypeGallery
ControllLineThicknessGallery
GroupMacroTools
ReplicationOptionsMenu

imageMso
GroupRecords
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
TextChevronInverted

imageMso
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
TextCanDown
GroupHeaderFooterExcel
HeaderFooterHeaderGallery

imageMso
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
PivotTableStylesGallery
FormatAsTableGallery
TableStylesGalleryExcel
ConditionalFormattingsManage
ConditionalFormattingHighlightGreaterThan

imageMso
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
CoverPageInsertGallery
PageNumberFieldInsertGallery
WatermarkGallery
EquationInsertGallery
QuickTablesInsertGallery
QuickPartsInsertGallery
SharePointListsDiscardAllChanges

imageMso
SortRemoveAllSorts
GroupSmartArtLayouts
GroupSmartArtQuickStyles
GroupSmartArtChooseColor
GroupSmartArtCreateGraphic
GroupSmartArtReset
GroupSmartArtSize
ConditionalFormattingHighlightCellsMenu
ConditionalFormattingTopBottomMenu
SaveSelectionToQuickPartGallery
SaveSelectionToCoverPageGallery
SaveSelectionToEquationGallery
SaveSelectionToFooterGallery
SaveSelectionToHeaderGallery
SaveSelectionToPageNumberGallery
SaveSelectionToQuickTablesGallery
SaveSelectionToWaterMarkGallery
MenuView2
FormatCellsMenu
ConditionalFormattingClearMenu
FontColorCycle
CellsInsertSmart
CellsDeleteSmart
ReviewShowMarkupMenu
ObjectEditShapeMenu
PivotTableLayoutReportLayout
PivotTableLayoutShowInCompactForm
PivotTableLayoutShowInOutlineForm
PivotTableLayoutShowInTabularForm
SymbolInsertGallery
ControlChart
SpellingAccess

imageMso
ViewsDatasheetView
PivotTableClearMenu
MacroArguments
MacroShowAllActions
GroupControlsAccess
CreateTable
CreateTableTemplatesGallery
AccessTableContacts
AccessTableTasks
AccessTableIssues
AccessTableEvents
AccessTableAssets
CreateTableUsingSharePointListsGallery
AccessListContacts
AccessListTasks
AccessListIssues
AccessListEvents
AccessListAssets
AccessListCustom
AccessListCustomDatasheet
CreateForm
CreateFormSplitForm
CreateFormWithMultipleItems
CreateFormMoreFormsGallery
AccessFormWizard
CreateFormPivotChart
AccessFormPivotTable
AccessFormDatasheet
AccessFormModalDialog
CreateFormBlankForm
CreateReport
AccessReportMore

imageMso
CreateReportFromWizard
CreateLabels
CreateReportBlankReport
CreateOtherObjectsMenu
ControlTitle
ControlLogo
ShowMargins
SlideMasterContentPlaceholderInsert
SlideMasterTextPlaceholderInsert
SlideMasterChartPlaceholderInsert
SlideMasterTablePlaceholderInsert
SlideMasterDiagramPlaceholderInsert
SlideMasterMediaPlaceholderInsert
SlideMasterClipArtPlaceholderInsert
SlideMasterVerticalTextPlaceholderInsert
SlideMasterShowFooters
CellStylesGallery
CellStyleNew
CellStylesMerge
TableStylesGallery
TableStyleNew
TableStyleClear
FilePublishExcelServices
PivotTableOlapConvertToFormulas
PivotTableLayoutBlankRows
TableStyleFirstColumn
TableStyleLastColumn
TableStyleBandedRows
TableStyleBandedColumns
TableStyleRowHeaders
TableStyleColumnHeaders
TableSummarizeWithPivot

imageMso
DisplayRuler
GetExternalDataFromAccess
GetExternalDataFromOtherSources
GetExternalDataExistingConnections
GroupThemesExcel
PivotPlusMinusButtonsShowHide
FileSaveAsPdfOrXps
FormattingDataType
FormattingRequiredField
FormattingFormat
ApplyCurrencyFormat
ApplyPercentageFormat
ApplyCommaFormat
FormattingIncreaseDecimals
FormattingDecreaseDecimals
BusinessCardInsertMenu
RecentFileList
DataGraphicIconSet
SearchLibraries
MoreControlsDialog
GroupCode
GroupRestrictions
GroupXml
PageScaleToFitWidth
PageScaleToFitHeight
SlideTransitionApplyToAll
AnimationPreview
XmlStructure
XmlSchema
XmlExpansionPacksWord
GroupCaptions
GroupIndex

imageMso
GroupTableOfAuthorities
GroupEditing
SelectMenuExcel
PrintTitles
NameUseInFormula
CalculationOptionsMenu
XmlSource
GroupClipboard
GroupInsertTables
GroupInsertLinks
GroupInsertSymbols
GroupInsertBarcode
GroupCalculation
BordersGallery
BordersMoreDialog
PageScaleToFitOptionsDialog
PageSizeGallery
AnimationTransitionSoundGallery
ObjectPictureFill
TextWrappingMenu
RecordsAddFromOutlook
RecordsSaveAsOutlookContact
DatasheetNewField
PivotTableGroupSelection
PivotTableGroupField
SlideShowFromBeginning
WindowSwitchWindowsMenuWord
WindowSwitchWindowsMenuPowerPoint
WindowSwitchWindowsMenuExcel
ShapeFillMoreGradientsDialog
ShadowOptionsDialog
BuildingBlocksOrganizer

imageMso
TableSharePointListsModifyColumnsAndSettings
TableListAlertMe
TableSharePointListsModifyWorkflow
TableListPermissions
TableSharePointListsRefreshList
CreateQueryInDesignView
ControlAttachment
GroupFieldsAndColumns
DataOptionsMenu
FormattingUnique
RecordsRefreshMenu
GroupLayoutShowHide
GroupQueryShowHide
GroupMacroShowHide
GroupPosition
GroupAnalyze
GroupControlLayout
GroupRelationshipsTools
GroupTableDesignTools
GroupTableRows
GroupSharePointList
GroupPivotTableFilterAndSort
GroupPivotTableToolsAccess
GroupPivotTableShowHideAccess
GroupPivotTableSelections
GroupPivotTableDataAccess
GroupPivotTableActiveFieldAccess
GroupPivotChartFilterAndSort
GroupPivotChartTools
GroupPivotChartShowHide
GroupPivotChartDataAccess
GroupPivotChartActiveFieldAccess

imageMso
GroupPivotChartType
ControlMarginsGallery
ControlPaddingGallery
AutoFormatGallery
PivotMoveField
PivotChartSortByTotalMenu
SharePointListsDiscardChangesMenu
ReviewCompareMenu
ReviewCompareTwoVersions
ReviewCombineRevisions
ReviewCompareMajorVersion
ReviewCompareLastVersion
ReviewCompareSpecificVersion
PropertyInsert
ObjectsAlignLeftSmart
ObjectsAlignRightSmart
ObjectsAlignTopSmart
ObjectsAlignBottomSmart
ObjectsAlignCenterHorizontalSmart
ObjectsAlignMiddleVerticalSmart
AlignDistributeHorizontally
AlignDistributeVertically
GroupProtect
IndexUpdate
MailMergeFinishAndMergeMenu
HyphenationMenu
MailMergeRules
SlideShowCustomMenu
SlideShowResolutionGallery
SlideShowUseRehearsedTimings
SlideShowShowPresentationOnGallery
GroupMonitors

imageMso
MailMergeStartMailMergeMenu
MailMergeStartLetters
MailMergeStartEmail
MailMergeStartEnvelopes
MailMergeStartLabels
MailMergeStartDirectory
MailMergeClearMergeType
MailMergeSelectRecipients
TableOfContentsAddTextGallery
FunctionsRecentlyUsedtInsertGallery
FunctionsFinancialInsertGallery
FunctionsDateTimeInsertGallery
FunctionsMathTrigInsertGallery
FunctionsTextInsertGallery
FunctionsLogicalInsertGallery
FunctionsStatisticalInsertGallery
FunctionsLookupReferenceInsertGallery
FunctionsInformationInsertGallery
GroupTableStyleOptions
GroupTableExternalData
GroupEditingExcel
BorderColorPicker
TranslationToolTip
FileCompatibilityChecker
FollowUpComposeMenu
ThemeFontsGallery
ThemeEffectsGallery
PrintColumns
PrintDataOnly
GroupControlAlignment
RmsNavigationBarHome
FileProperties

imageMso
ForwardAttach
BackAttach
ManageAttachments
GroupPrintPreviewPrint
GroupPrintPreviewPreview
GroupPlay
GroupMovieOptions
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

imageMso
RejectAndAdvance
FileCreateDocumentWorkspace
FileSaveToDocumentManagementServer
FileDocumentManagementInformation
ScreenNavigatorBack
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

imageMso
GroupMasterLayout
GroupColorModeSetting
ShadowStyleGalleryClassic
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

imageMso
GroupTableSize
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
OmsEmoticonInsertGallery

imageMso
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
RssShareThisFeed
OpenInBrowserOutlook
InsertAlignmentTab

imageMso
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
GroupShadowEffects
Group3DEffects
WordArtStylesGalleryClassic
WordArtInsertGalleryClassic
TableInsertGallery

imageMso
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
CDAudioStopTrack
CDAudioStartTime
CDAudioStopTime
SlideOrientationGallery
TextHighlightColorPicker
PageColorPicker
GoToHeader

imageMso
GoToFooter
UnderlineGallery
UnderlineColorPicker
TextDirectionGalleryWord
GroupAddInsMenuCommands
GroupAddInsToolbarCommands
ViewFreezePanelsGallery
GroupInk
GroupInkPens
GroupInkClose
InkBallpointPen
InkFeltTipPen
InkHighlighter
PhoneticGuideMenu
PageScaleToFitScale
GroupBorder
PictureRecolorGalleryWord
_3DSurfaceMaterialGalleryClassic
_3DExtrusionDepthGalleryClassic
GroupHeaderFooter
GroupPageLayoutSetup
GroupPageBackground
FunctionsCubeInsertGallery
FunctionsEngineeringInsertGallery
ThemeSaveCurrentPowerPoint
ThemeSaveCurrent
TextAlignGallery
ThemesGallery
CharacterSpacingGallery
ChartFormatDataSeries
ChartFormatDataLabels
ChartResetToMatchStyle

imageMso
ChartSeriesTypeChange
Chart3DView
ChartFormatDataPoint
ChartFormatDataLabel
ChartFormatAxis
ChartFormatGridlines
ChartFormatDisplayUnit
ChartFormatLegendEntry
ChartFormatChartArea
ChartFormatLegend
ChartFormatErrorBars
ChartFormatUpBars
ChartFormatDownBars
ChartFormatHighLowLine
ChartFormatDropLines
ChartFormatTrendline
ChartFormatTrendlineLabel
ChartFormatSeriesLine
ChartFormatDataTable
ChartFormatAxisTitle
ChartFormatChartTitle
ChartFormatFloor
ChartFormatSideWall
ChartFormatBackWall
ChartFormatWalls
ChartFormatPlotArea
ObjectSizeAndPositionDialog
ObjectSizeAndPropertiesDialog
ObjectSizeDialog
ShapeConvertToFreeform
AutoTextGallery
TextBoxInsertGallery

imageMso
PageNumbersInHeaderInsertGallery
PageNumbersInFooterInsertGallery
PageNumbersInMarginsInsertGallery
SaveSelectionToAutoTextGallery
SaveSelectionToTextBoxGallery
BuildingBlocksCreateLayout
SaveSelectionToPageNumberTop
SaveSelectionToPageNumberBottom
SaveSelectionToPageNumberMargin
BuildingBlocksSaveCoverPage
BuildingBlocksSaveHeader
BuildingBlocksSaveFooter
BuildingBlocksSavePageNumTop
BuildingBlocksSavePageNumBottom
BuildingBlocksSavePageNum
HeaderFooterEditHeader
TableStyleBandedRowsWord
TableStyleBandedColumnsWord
MSWordApplyTableStyle
TableStyleModify
TableStylesGalleryWord
ReviewViewChangesInTheSourceDocument
GroupThemesWord
OfficeDiagnostics
WindowsSwitch
LayoutOptionsDialog
DrawingObjectFormatDialog
FontAlternateFillBackColorPicker
GroupGroupingAndTotals
GroupFindAccess
GroupSchemaTools
GroupAdpDiagramShowHide

imageMso
GroupAdpDiagramLayout
GroupAdpQueryTools
GroupAdpOutputOperations
GroupAdpQueryType
GroupAdpSqlStatementDesignTools
ConvertToSmartArt
ConvertToSmartArtMoreSmartArtGraphicsDialog
ReflectionGallery
PictureRecolorGallery
SmartArtPromote
SmartArtDemote
TableEffectsCellBevelGallery
GroupPivotChartShowOrHide
GroupPivotChartData
GroupPivotChartActiveField
PivotChartRefresh
PivotChartFilterShow
TableBackgroundGallery
_3DPerspectiveIncrease
_3DPerspectiveDecrease
ChartSecondaryHorizontalAxisTitle
ChartSecondaryVerticalAxisTitle
ChartSecondaryHorizontalGridlines
ChartSecondaryVerticalGridlines
ChartSecondaryHorizontalAxis
ChartSecondaryVerticalAxis
ReadingModeToPrintView
GroupAddInsCustomToolbars
MoviePlayAutomatically
ReviewDeleteCommentsMenu
ConditionalFormattingMenu
GroupPersonalInfo

imageMso
GroupMessageOptions
GroupAttach
GroupNames
GroupRespond
GroupMembers
GroupContactOptions
GroupJunkEmail
ForwardMenu
JunkEmailSafeListsMenu
UseVotingButtonsMenu
RecentlyUsedFolder
ArrangeWindowsDialog
GroupForm
MakeSameSizeMenu
CategorizeMenu
ReplyAllWithInstantMessage
ObjectBringToFrontMenu
ObjectSendToBackMenu
ObjectsGroupMenu
SignatureLineInsertMenu
FileSaveAsWord97_2003
ViewDisplayInColor
HyperlinkFlyoutAnchor
FileSaveAsPowerPoint97_2003
SelectionPaneHidden
PivotFieldList
FilePackageAndSign
TotalsMenu
EnglishWritingAssistant
TableOfContentsGallery
SaveSelectionToTableOfContentsGallery
BuildingBlocksCreateTableOfFigures

imageMso
BuildingBlocksCreateTableOfAuthorities
BuildingBlocksCreateIndex
BuildingBlocksSaveTableOfContents
BuildingBlocksSaveTableOfFigures
BuildingBlocksSaveTableOfAuthorities
BuildingBlocksSaveIndex
FileSaveAsExcel97_2003
TextBoxInsertExcel
TextBoxStyleGallery
TableColumnWidth
TableRowHeight
TextBoxPositionGallery
FilePermissionRestrictMenu
GroupEnvelopeLabelCreate
ReviewInkCommentNew
InsertCellstMenu
PivotTableInsertMenu
OutlineGroupMenu
OutlineUngroupMenu
FormulaMoreFunctionsMenu
WordInsertTableOfContents2
ReadingViewTrackChanges
ReadingViewShowCommentsAndChangesMenu
ReadingViewChangeInkPenMenu
DocumentPanelTemplate
GroupModify
ObjectSizeDialogClassic
ViewGridlinesToggleExcel
TextBoxInsertVerticalWord
BevelShapeGallery
_3DBevelPictureTopGallery
_3DMaterialMixed

imageMso
_3DMaterialPlastic
_3DMaterialMetal
BibliographyAddNewPlaceholder
TableStyleClearTable
GroupTable
GroupTableCellSize
TableEffectsMenu
EditLinks
PositionAbsoluteMarks
QuerySplitMenu
FileSaveAsMenuAccess
FileServerMenu
GlowColorPicker
RecolorColorPicker
GlowColorMoreColorsDialog
PictureRecolorMoreColorsDialog
ShapesDuplicate
CloseDocument
WatermarkRemove
CoverPageRemove
HeaderFooterRemoveHeaderWord
HeaderFooterRemoveFooterWord
PageNumbersRemove
TableOfContentsRemove
SmartArtAddBullet
TableWidth
TableHeight
SlideMasterPicturePlaceholderInsert
SlideMasterInsertPlaceholderMenu
RemoveCitation
EditCitation
TableBorderColorMoreColorsDialog

imageMso
TableFillColorMoreColorsDialog
PictureChange
GroupWordArtStyles
TextFillColorPicker
TextOutlineColorPicker
TextOutlineColorMoreColorsDialog
TextEffectsMenu
TextStylesGallery
WordArtClear
TextPictureFill
TextFillGradientGallery
TextFillMoreGradientsDialog
TextFillTextureGallery
TextOutlineDashesGallery
TextOutlineMoreLinesDialog
TextOutlineWeightGallery
TextEffectShadowGallery
TextEffectsMoreShadowsDialog
TextEffectsBevelMore3DOptionsDialog
TextEffects3DRotationGallery
TextEffects3DRotationOptionsDialog
TextEffectGlowGallery
TextGlowColorPicker
TextGlowColorMoreColorsDialog
TextReflectionGallery
ShapeEffectsMenu
NewTableStyle2
NewPivotTableStyle2
ShapeConnectorStyleMenu
PivotTableSubtotalsDoNotShow
PivotTableSubtotalsOnBottom
PivotTableSubtotalsOnTop

imageMso
PivotTableGrandTotalsOffForRowsAndColumns
PivotTableGrandTotalsOnForRowsAndColumns
PivotTableGrandTotalsOnForRowsOnly
PivotTableGrandTotalsOnForColumnsOnly
PivotTableBlankRowsInsert
PivotTableBlankRowsRemove
ShowMessagePage
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

imageMso
ItemProperties
PermissionRestrictMenu
MoveItem
ShowSharePage
EditBusinessCard
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

imageMso
GroupSend
IMMenu
UpgradeDocument
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

imageMso
ContentControlPicture
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
BibliographyGallery

imageMso
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
GroupPictureSizeClassic
GroupPictureToolsClassic
GalleryAllShapesAndCanvas

imageMso
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
GroupCommentShapes
GroupCommentPosition
TripaneViewMode
PrintOptionsMenu
FileMenuSendHeader

imageMso
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
EquationFractionGallery
EquationRadicalGallery
EquationLargeOperatorGallery
EquationDelimiterGallery
EquationScriptGallery
EquationFunctionGallery
EquationAccentGallery

imageMso
EquationLimitGallery
EquationOperatorGallery
EquationMatrixGallery
EquationOptions
GroupEquationTools
GroupEquationSymbols
GroupEquationStructures
EquationOptionsMenu
GroupBasicText
GroupInclude
GroupFields
GroupMessageFormat
GroupTracking
GroupMoreOptions
GroupCommunicate
GroupTimer
SaveSentItemsMenu
DelayDeliveryOutlook
DirectRepliesTo
SaveSentItemRecentlyUsedFolder
SaveSentItemOtherFolder
FindRelatedMenu
OtherActionsMenu
EncodingMenu
Drawing1GalleryRotateObject
MailMergeMergeFieldInsertMenu
ReviewDeleteAllMarkupInPresentation
PasteMenu
GroupPictureStyles
PictureStylesGallery
GroupInsertChartsExcel
ChartTypeColumnInsertGallery

imageMso
ChartTypeLineInsertGallery
ChartTypePieInsertGallery
ChartTypeBarInsertGallery
ChartTypeAreaInsertGallery
ChartTypeXYScatterInsertGallery
ChartTypeOtherInsertGallery
ChartTypeAllInsertDialog
PivotChartClearMenu
ReviewAcceptChangeAndMoveToNext
ReviewRejectChangeAndMoveToNext
PictureEffectsShadowGallery
PictureEffectsGlowGallery
PictureEffectsSoftEdgesGallery
PictureReflectionGallery
PictureReflectionGalleryItem
PictureRotationGallery
ReadingViewShowOriginalOrFinalDocument
InterconnectDeleteCard
InkToolsClose
GroupChineseTranslation
CloseMasterView
SheetRowsInsert
SheetColumnsInsert
SheetRowsDelete
SheetColumnsDelete
JustifyVerticalFlyoutAnchor
JustifyLowVertical
JustifyMediumVertical
JustifyHighVertical
LineSpacingMenu
WordOpenParaAbove
WordCloseParaAbove

imageMso
WordOpenParaBelow
WordCloseParaBelow
AdpNewTable
ViewsSwitchToDefaultView
GroupDocumentViews
GroupViewShowHide
GroupWindow
GroupWorkbookViews
ViewGridlines
ViewHeadings
GroupPresentationViews
GroupColorGrayscale
RefreshAllMenu
DataValidationMenu
GroupViews
GroupViewsShowHide
GroupViewZoom
GroupWindowAccess
FPTableAutoFormat
FPTable
PageViewMenu
ShowTimeZones
ViewOnlineConnection
CLViewDialogHelpID
FileDocumentEncrypt
WordArtFormatDialog
MoreTextureOptions
TextFillColorMoreColorsDialog
ZoomFitToWindow
ZoomTwoPages
BorderTopNoToggle
BorderBottomNoToggle

imageMso
BorderLeftNoToggle
BorderRightNoToggle
WindowSplitToggle
QuickPartsInsertFromOnline
JotShapeRectangle
JotShapeEllipse
JotShapeParallelogram
JotShapeDiamond
JotSendPdf
ProjectTaskDrivers
ProjectStatusReports
ProjectRecalcChangeHighlighting
ProjectAdvancedDesktopReporting
ProjectManageDeliverables
ProjectManageDependenciesOnDeliverables
JotLineColor
AddInsMenu
GroupPrintPreviewPageSetup
GroupPrintPageBackground
ShowRuler
ViewDirectionMenu
JustifyThaiVertical
FileEmailAsPdfEmailAttachment
FileEmailAsXpsEmailAttachment
FontFillBackColorPicker
GroupArrangeOutlook
GroupTemplates
SelectMenuAccess
GoToMenuAccess
GroupQueryResults
GroupAdminister
GroupTableDesignShowHide

imageMso
GroupPrintPreviewData
GroupPrintPreviewClosePreview
GroupFormatting
GroupQueryClose
GroupMacroClose
ObjectsGroupMenuOutlook
RegionLayoutMenu
ResponsesMenu
GroupOpen
SpellingMenu
GroupPrintPreviewZoom
PictureEffectsMenu
PictureShapeGallery
PictureBorderColorPickerClassic
GroupInsertShapes
PhotoAlbumInsertMenu
GroupChartBackground
GroupChartAnalysis
ZoomToSelection
FileNewBlogPost
GroupBlogPublish
BlogPublishMenu
BlogPublish
BlogPublishDraft
BlogManageAccounts
GroupBlogProperties
BlogCategories
BlogCategoryInsert
BlogCategoriesRefresh
BlogOpenExisting
GroupBlogBasicText
GroupBlogInsertText

imageMso
DeleteSlideContextual
AutoSigInsertPictureFromFile
AutoSigWebInsertHyperlink
GroupFind
ViewPreviousItemMenu
ViewNextItemMenu
ContactLinkMenu
NewTableStyleWord
CacheListData
AccessRefreshAllLists
AccessOfflineLists
AccessOnlineLists
GridlinesExcel
UnmergeCells
MenuPublish
FileMenuPublishHeader
ChangeStylesMenu
GroupBlogInsertLinks
SharePointSiteRecycleBin
SubformMenu
LabelFontDialog
PrintDialogAccess
ForwardInForwardMenu
FileCompatibilityCheckerPowerPoint
SharingNone
AccessRecycleBin
FileCompatibilityCheckerWord
AccessRelinkLists
StopAnimation
ViewSideBySide
TableBackgroundPictureFill
GroupAttendeesMeetingSent

imageMso
ImexRunImport
ImexRunExport
GroupTools
GroupFormattingControls
GroupControlAlignmentLayout
GroupDesignGridlines
GroupFormattingGridlines
GroupFieldsTools
GroupFontAccess
GroupDatasheetRelationships
GroupPositionLayout
GroupControlPositionLayout
GroupControlSize
PrintPreviewZoomMenu
GroupMarginsAndPaddingControlLayout
RecordsDeleteColumn
ViewsAdpDiagramPrintPreview
UpgradePresentation
ChartShowDataContextualMenu
ChartSourceDataContextualMenu
FileSaveAsOtherFormats
FileSaveAsWordDocx
FileSaveAsWordDotx
FileSaveAsExcelXlsx
FileSaveAsExcelXlsxMacro
FileSaveAsExcelXlsb
FileSaveAsPowerPointPptx
FileSaveAsPowerPointPpsx
LockCell
PasteAsPictureMenu
GroupPivotActions
PivotTableSelectFlyout

imageMso
ZoomCurrent100
Drawing1ColorPickerLineStyleWordArt
Drawing1ColorPickerFillWordArt
FontDialogPowerPoint
PowerPointParagraphDialog
PowerPointPageSetup
GroupDrawing2
ShapeQuickStylesHome
GalleryAllShapesAndTextboxes
TableTextStylesGallery
GroupTextStylesTable
UpdateBibliography
ExportToAccess
BizBarPublishToSharePoint
BlogInsertCategories
NewMailMessageNumbered
NewAppointmentNumbered
NewMeetingRequestNumbered
NewContactNumbered
NewDistributionListNumbered
NewTaskNumbered
NewNoteNumbered
FileMenuSaveAsHeaderOutlook
FileMenuPrintHeaderOutlook
GroupRecover
SmartArtInsertBulletRTL
NewContactMenu
GroupTableLayout
ObjectsArrangeMenu
TextFillMoreTextures
FileMenuPrintHeaderAccess
ConvertDatabaseFormat

imageMso
PivotShowOnlyTheTopMenu
PivotShowOnlyTheBottomMenu
GroupMacros
PlayMacro
MenuMacros
SharingOpenWssDocumentList
SharingOpenWssDiscussionList
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

imageMso
ChartFormatLeaderLines
ThemeBrowseForThemes
FilePublishToSharePoint
FileCheckOutDiscard
GroupBlogSymbols

4 Appendix B: Full XML Schemas

4.1 <http://schemas.microsoft.com/office/2006/01/customui> Schema

```
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0"
  targetNamespace="http://schemas.microsoft.com/office/2006/01/customui"
  xmlns="http://schemas.microsoft.com/office/2006/01/customui"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:annotation>
    <xsd:documentation>
      -----
      Schema definition for Ribbon Extensibility
      -----
    </xsd:documentation>
  </xsd:annotation>
  <xsd:annotation>
    <xsd:documentation>
      -----
      Attribute types
      -----
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType name="ST_QID">
    <xsd:annotation>
      <xsd:documentation>
        The qualified ID of a control.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:QName">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_ID">
    <xsd:annotation>
      <xsd:documentation>
        The custom ID of a control.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:NCName">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_UniqueID">
    <xsd:annotation>
      <xsd:documentation>
        A unique ID.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:ID">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_Delegate">
    <xsd:annotation>
      <xsd:documentation>
        A callback. Callbacks are used to provide status, update properties or perform
        actions.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```

</xsd:simpleType>
<xsd:simpleType name="ST_StringLength">
  <xsd:annotation>
    <xsd:documentation>
      A numeric argument for maximum string length in controls.
      String length is limited to 1024 characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="1024"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_GalleryRowColumnCount">
  <xsd:annotation>
    <xsd:documentation>
      A numeric argument for the maximum number of rows or columns in galleries.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="1024"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_GalleryItemWidthHeight">
  <xsd:annotation>
    <xsd:documentation>
      A numeric argument for the maximum width or height of a gallery item.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="4096"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST String">
  <xsd:annotation>
    <xsd:documentation>
      A string argument.
      String length is limited to 1024 characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST LongString">
  <xsd:annotation>
    <xsd:documentation>
      A long string argument.
      String length is limited to 4096 characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="4096"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST Uri">
  <xsd:annotation>
    <xsd:documentation>
      A string argument for a path to a file or a resource.
      String length is limited to 1024 characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">

```

```

    <xsd:minLength value="1"/>
    <xsd:maxLength value="1024"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ST_Size">
  <xsd:annotation>
    <xsd:documentation>
      The size of a button. Values are "normal" or "large."
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="normal"/>
    <xsd:enumeration value="large"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_ItemSize">
  <xsd:annotation>
    <xsd:documentation>
      The size of items in a menu. Values are "normal" or "large."
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="normal"/>
    <xsd:enumeration value="large"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_BoxStyle">
  <xsd:annotation>
    <xsd:documentation>
      A box control. Values are "horizontal" or "vertical."
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="horizontal"/>
    <xsd:enumeration value="vertical"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_Keytip">
  <xsd:annotation>
    <xsd:documentation>
      A keytip. Value is 1-3 alphanumeric characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:token">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="3"/>
    <xsd:whiteSpace value="collapse"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:annotation>
  <xsd:documentation>
    -----
    Attributes
    -----
  </xsd:documentation>
</xsd:annotation>
<xsd:attributeGroup name="AG_IDCustom">
  <xsd:annotation>
    <xsd:documentation>
      Attributes for a custom control ID.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="id" type="ST_UniqueID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The ID of a custom UI element. IDs must be unique.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>

```

```

    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="idQ" type="ST_QID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        A qualified control ID. Qualified IDs allow different add-ins to modify the same
        custom group, tab, or menu.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_IDMso">
  <xsd:annotation>
    <xsd:documentation>
      The ID of a built-in control.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="idMso" type="ST ID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The ID of a built-in control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Tag">
  <xsd:annotation>
    <xsd:documentation>
      Custom data.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="tag" type="ST String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Custom data.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Title">
  <xsd:annotation>
    <xsd:documentation>
      The title of a menu.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="title" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The title of a menu.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getTitle" type="ST Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback for the title of a menu.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_IDAttributes">
  <xsd:annotation>
    <xsd:documentation>
      Attributes that specify control ID.
      One of id, idMso, or idQ must be specified to identify a control.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attributeGroup ref="AG_IDCustom"/>

```

```

<xsd:attributeGroup ref="AG_IDMso"/>
<xsd:attributeGroup ref="AG_Tag"/>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Image">
  <xsd:annotation>
    <xsd:documentation>

      The image or icon of a control.
      Image attributes are mutually exclusive - only one of
      "image", "imageMso", or "getImage" may be specified.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="image" type="ST_Uri" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        A custom image or icon.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="imageMso" type="ST_ID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The image of a built-in control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getImage" type="ST Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback for a custom image.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_CommonAttributes">
  <xsd:annotation>
    <xsd:documentation>
      Attributes that can be applied to all commands and controls.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attributeGroup ref="AG_Enabled"/>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_PositionAttributes">
  <xsd:annotation>
    <xsd:documentation>
      Attributes that set the position of an object relative to its siblings,
      such as the position of a control within a group or position of a tab relative to
other tabs.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="insertAfterMso" type="ST_ID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The ID of built-in control to be inserted after.
        Mutually exclusive with InsertBeforeMso, InsertAfterQ, InsertBeforeQ.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="insertBeforeMso" type="ST_ID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The ID of built-in control to be inserted before.
        Mutually exclusive with InsertAfterMso, InsertAfterQ, InsertBeforeQ.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="insertAfterQ" type="ST QID" use="optional" >
    <xsd:annotation>
      <xsd:documentation>

```

```

        The ID of control to be inserted after.
        Mutually exclusive with InsertAfterMso, InsertBeforeMso, InsertBeforeQ.
    </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="insertBeforeQ" type="ST_QID" use="optional" >
    <xsd:annotation>

        <xsd:documentation>
            The ID of control to be inserted before.
            Mutually exclusive with InsertAfterMso, InsertBeforeMso, InsertAfterQ.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG Enabled">
    <xsd:annotation>
        <xsd:documentation>
            Attributes for a control's enabled state.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="enabled" type="xsd:boolean" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Whether the control is enabled.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="getEnabled" type="ST_Delegate" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Callback that returns true if the control is enabled.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG Visible">
    <xsd:annotation>
        <xsd:documentation>
            Attributes for a control's visibility.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="visible" type="xsd:boolean" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Whether the control is visible.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="getVisible" type="ST_Delegate" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Callback that returns true if the control is visible.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Label">
    <xsd:annotation>
        <xsd:documentation>
            Attributes that set a control's label.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="label" type="ST_String" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Sets the label.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:attributeGroup>

```

```

</xsd:attribute>
<xsd:attribute name="getLabel" type="ST_Delegate" use="optional">
  <xsd:annotation>
    <xsd:documentation>
      Callback that sets the label.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Keytip">
  <xsd:annotation>
    <xsd:documentation>
      Attributes to set a control's keytip.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="keytip" type="ST_Keytip" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Sets the keytip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getKeytip" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback that sets the keytip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Screentip">
  <xsd:annotation>
    <xsd:documentation>
      Attributes for a control's screentip.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="screentip" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Sets the screentip, which appears on mouse hover.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getScreentip" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback that sets the screentip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="supertip" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Sets the supertip, a large screentip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getSupertip" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback that sets the supertip, a large screentip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Description">
  <xsd:annotation>
    <xsd:documentation>

```



```

    Attributes for the extended description of a control.
  </xsd:documentation>
</xsd:annotation>
<xsd:attribute name="description" type="ST_LongString" use="optional">
  <xsd:documentation>
    <xsd:documentation>
      Sets the extended description of the control, which appears in menus with itemSize
set to large.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getDescription" type="ST_Delegate" use="optional">
  <xsd:documentation>
    <xsd:documentation>
      Callback that sets the description.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_UIAttributes">
  <xsd:documentation>
    <xsd:documentation>
      Attributes that can be applied to all controls.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
<xsd:attributeGroup ref="AG_CommonAttributes"/>
<xsd:attributeGroup ref="AG_Label"/>
<xsd:attributeGroup ref="AG_PositionAttributes"/>
<xsd:attributeGroup ref="AG_Visible"/>
<xsd:attributeGroup ref="AG_Keytip"/>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_ItemAttributes">
  <xsd:documentation>
    <xsd:documentation>
      Common attributes that can be applied to controls and groups.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
<xsd:attributeGroup ref="AG_Image"/>
<xsd:attributeGroup ref="AG_Screentip"/>
<xsd:attributeGroup ref="AG_UIAttributes"/>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_ControlAttributes">
  <xsd:documentation>
    <xsd:documentation>
      Attributes applied to controls.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
<xsd:attributeGroup ref="AG_ItemAttributes"/>
<xsd:attribute name="showLabel" type="xsd:boolean" use="optional">
  <xsd:documentation>
    <xsd:documentation>
      Whether to show a control's label.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getShowLabel" type="ST_Delegate" use="optional">
  <xsd:documentation>
    <xsd:documentation>
      Callback for whether to show a control's label.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="showImage" type="xsd:boolean" use="optional">
  <xsd:documentation>
    <xsd:documentation>
      Whether to show a control's image.
    </xsd:documentation>
  </xsd:documentation>
</xsd:annotation>
</xsd:attribute>

```

```

</xsd:attribute>
<xsd:attribute name="getShowImage" type="ST_Delegate" use="optional">
  <xsd:annotation>
    <xsd:documentation>
      Callback for whether to show a control's image.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_Action">

  <xsd:annotation>
    <xsd:documentation>
      Callback fired on user action.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="onAction" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback fired on user action (for example, a button press).
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_SizeAttributes">
  <xsd:annotation>
    <xsd:documentation>
      A size attribute.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="size" type="ST_Size" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The size of a control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getSize" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback for a control's size.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_DropDownAttributes">
  <xsd:annotation>
    <xsd:documentation>
      Common attributes for controls with dropdowns (such as comboBox, gallery, or
      dropDown).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="showItemImage" type="xsd:boolean" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Whether item images are shown in the dropdown.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getItemCount" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback for the number of items in the dropdown.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getItemLabel" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>

```

```

        Callback for an item's label.
    </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getItemScreentip" type="ST_Delegate" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            Callback for an item's screentip.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>

<xsd:attribute name="getItemSupertip" type="ST_Delegate" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            Callback for an item's supertip.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getItemImage" type="ST_Delegate" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            Callback for an item's image.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getItemID" type="ST_Delegate" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            Callback for an item's ID.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="sizeString" type="ST_String" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            A representative string that sets the control's width.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_GetContentAttributes">
    <xsd:annotation>
        <xsd:documentation>
            Attributes for dynamic controls that support the getContent callback.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="getContent" type="ST_Delegate" use="required">
        <xsd:annotation>
            <xsd:documentation>
                Callback that returns the dynamic content for this control.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:attributeGroup>
<xsd:attributeGroup name="AG_DynamicContentAttributes">
    <xsd:annotation>
        <xsd:documentation>
            Attributes for controls that support dynamic content.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="invalidateContentOnDrop" type="xsd:boolean" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                Whether to fire callback for dynamic content each time the control is dropped.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:attributeGroup>

```

```

<xsd:annotation>
  <xsd:documentation>
    -----
    Global settings
    -----
  </xsd:documentation>
</xsd:annotation>
<xsd:complexType name="CT_Command" mixed="false">
  <xsd:annotation>
    <xsd:documentation>
      Attribute overrides for all controls with specified built-in ID.

      For example
      <command idMso="Print" enabled="false">
      disables all instances of the Print button.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attributeGroup ref="AG_Action"/>
  <xsd:attributeGroup ref="AG_Enabled"/>
  <xsd:attributeGroup ref="AG_IDMso"/>
</xsd:complexType>

<xsd:annotation>
  <xsd:documentation>
    -----
    Controls
    -----
  </xsd:documentation>
</xsd:annotation>
<xsd:complexType name="CT_ControlBase">
  <xsd:annotation>
    <xsd:documentation>
      Base control type.
      Doesn't define ID attributes.
      Abstract type, not to be used directly.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attributeGroup ref="AG_ControlAttributes"/>
</xsd:complexType>
<xsd:complexType name="CT_Control">
  <xsd:annotation>
    <xsd:documentation>
      A type of control that can be used to
      enable, disable, or clone an existing control.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_ControlBase">
      <xsd:attributeGroup ref="AG_IDAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_ControlCloneRegular">
  <xsd:annotation>
    <xsd:documentation>
      A clone of a built-in control.
      Only the most common attributes can be applied here; to set
      control-specific properties the actual control type must be specified.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="CT_Control">
      <xsd:attribute name="id" use="prohibited">
        <xsd:annotation>
          <xsd:documentation>
            Custom controls can't be cloned.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:attribute>
    </xsd:restriction>
</xsd:complexType>
</xsd:complexType>
<xsd:complexType name="CT_ControlClone">
    <xsd:annotation>
        <xsd:documentation>
            A clone of built-in control that can be sized.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType base="CT_Button">
        <xsd:attribute name="id" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    Custom controls can't be cloned.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="onAction" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    OnAction does not apply to 'control'.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
    </xsd:restriction>
</xsd:complexType>
</xsd:complexType>
<xsd:complexType name="CT_LabelControl">
    <xsd:annotation>
        <xsd:documentation>
            Shows text and/or icon but can't have any associated actions.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType base="CT_Control">
        <xsd:attribute name="image" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    Image does not apply to labelControl.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="imageMso" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    ImageMso does not apply to labelControl.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="getImage" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    GetImage does not apply to labelControl.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="keytip" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>
                    Keytip does not apply to labelControl.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="getKeytip" use="prohibited">
            <xsd:annotation>
                <xsd:documentation>

```

```

        GetKeyTip does not apply to labelControl.
    </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="showImage" use="prohibited">
    <xsd:annotation>
        <xsd:documentation>
            ShowImage does not apply to labelControl.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="getShowImage" use="prohibited">
    <xsd:annotation>
        <xsd:documentation>
            GetShowImage does not apply to labelControl.
        </xsd:documentation>
    </xsd:annotation>
</xsd:restriction>
</xsd:complexType>
<xsd:complexType name="CT_ButtonRegular">
    <xsd:annotation>
        <xsd:documentation>
            A fixed-size button.
            Size of this button is determined by its container such as a menu.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_Control">
            <xsd:attributeGroup ref="AG_Action"/>
            <xsd:attributeGroup ref="AG_Enabled"/>
            <xsd:attributeGroup ref="AG_Description"/>
            <xsd:attributeGroup ref="AG_Image"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Button">
    <xsd:annotation>
        <xsd:documentation>
            A push-type button.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ButtonRegular">
            <xsd:attributeGroup ref="AG_SizeAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_VisibleButton">
    <xsd:annotation>
        <xsd:documentation>
            A button which is always visible.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:restriction base="CT_ButtonRegular">
            <xsd:attribute name="visible" use="prohibited">
                <xsd:annotation>
                    <xsd:documentation>
                        Visible does not apply to these buttons.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="getVisible" use="prohibited">
                <xsd:annotation>
                    <xsd:documentation>
                        GetVisible does not apply to these buttons.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_ToggleButtonRegular">
    <xsd:annotation>
        <xsd:documentation>
            A fixed-size button with an on/off state like the 'Bold' button.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ButtonRegular">
            <xsd:attribute name="getPressed" type="ST_Delegate" use="optional">

                <xsd:annotation>
                    <xsd:documentation>
                        Callback for whether the button is pressed.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_ToggleButton">
    <xsd:annotation>
        <xsd:documentation>
            A button with an on/off state that can be sized.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ToggleButtonRegular">
            <xsd:attributeGroup ref="AG_SizeAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_VisibleToggleButton">
    <xsd:annotation>
        <xsd:documentation>
            A toggleButton which is always visible.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:restriction base="CT_ToggleButtonRegular">
            <xsd:attribute name="visible" use="prohibited">
                <xsd:annotation>
                    <xsd:documentation>
                        Visible does not apply to these buttons.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="getVisible" use="prohibited">
                <xsd:annotation>
                    <xsd:documentation>
                        GetVisible does not apply to these buttons.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_CheckBox">
    <xsd:annotation>
        <xsd:documentation>
            A check box.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>

```

```

<xsd:restriction base="CT_ToggleButtonRegular">
  <xsd:attribute name="image" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        Image does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="imageMso" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        ImageMso does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="showImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        ShowImage does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getShowImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetShowImage does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="showLabel" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        ShowLabel does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getShowLabel" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetShowLabel does not apply to checkBox.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_EditBox">
  <xsd:annotation>
    <xsd:documentation>
      An editBox.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_Control">
      <xsd:attributeGroup ref="AG_Enabled"/>
      <xsd:attributeGroup ref="AG_Image"/>
      <xsd:attribute name="maxLength" type="ST_StringLength" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            The maximum number of characters the user may enter.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```



```

    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getText" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback for to populate text in the edit box before the user edits it.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="onChange" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback that fires when the user changes the editBox content.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="sizeString" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        A representative string that sets the control's width.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Item">
  <xsd:annotation>
    <xsd:documentation>
      An item in a dropdown-type control.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="id" type="ST_UniqueID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The ID of an item. Items cannot use idMso or idQ.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="label" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        An item label.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="image" type="ST Uri" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        An item image.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="imageMso" type="ST_ID" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        A built-in image.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="screentip" type="ST String" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        The screentip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="supertip" type="ST_String" use="optional">

```

```

    <xsd:annotation>
      <xsd:documentation>
        The supertip.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="CT_ComboBox">
  <xsd:annotation>
    <xsd:documentation>
      A comboBox control.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_EditBox">
      <xsd:sequence>
        <xsd:element name="item" type="CT Item" minOccurs="0" maxOccurs="1000">
          <xsd:annotation>
            <xsd:documentation>
              An item in a comboBox.
              When selected, the label of the item becomes text content of the comboBox.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attributeGroup ref="AG DropDownAttributes"/>
      <xsd:attributeGroup ref="AG_DynamicContentAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_DropDownRegular">
  <xsd:annotation>
    <xsd:documentation>
      A dropdown-type control with a fixed size.
      Contains items followed by buttons.
      OnAction reports the selected item.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_Control">
      <xsd:sequence>
        <xsd:element name="item" type="CT_Item" minOccurs="0" maxOccurs="1000">
          <xsd:annotation>
            <xsd:documentation>
              item in the dropdown.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="button" type="CT ButtonRegular" minOccurs="0" maxOccurs="16">
          <xsd:annotation>
            <xsd:documentation>
              Button in footer of dropdown.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attributeGroup ref="AG_Action"/>
      <xsd:attributeGroup ref="AG_Enabled"/>
      <xsd:attributeGroup ref="AG_Image"/>
      <xsd:attributeGroup ref="AG_DropDownAttributes"/>
      <xsd:attribute name="getSelectedItemID" type="ST Delegate" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Callback that returns the ID of currently selected item.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="getSelectedItemIndex" type="ST_Delegate" use="optional">

```

```

        <xsd:annotation>
          <xsd:documentation>
            Callback that returns the index of currently selected item.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="showItemLabel" type="xsd:boolean" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Whether the label is shown or hidden on dropdown items.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexType>
</xsd:complexType>
<xsd:complexType name="CT GalleryRegular">
  <xsd:annotation>
    <xsd:documentation>
      A dropdown grid control that can be sized.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT DropDownRegular">
      <xsd:attributeGroup ref="AG_Description"/>
      <xsd:attributeGroup ref="AG_DynamicContentAttributes"/>
      <xsd:attribute name="columns" type="ST_GalleryRowColumnCount" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Number of columns in dropdown gallery.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="rows" type="ST_GalleryRowColumnCount" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Number of rows in dropdown gallery.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="itemWidth" type="ST_GalleryItemWidthHeight" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Item width in pixels.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="itemHeight" type="ST_GalleryItemWidthHeight" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Item height in pixels.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="getItemWidth" type="ST_Delegate" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Callback that returns the item width.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="getItemHeight" type="ST_Delegate" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            Callback that returns the item height.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

    <xsd:attribute name="showItemLabel" type="xsd:boolean" use="optional">
      <xsd:annotation>
        <xsd:documentation>
          Whether the label is shown or hidden on gallery items.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Gallery">
  <xsd:annotation>
    <xsd:documentation>
      A gallery control.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_GalleryRegular">
      <xsd:attributeGroup ref="AG_SizeAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:group name="EG_MenuControlsBase">
  <xsd:annotation>
    <xsd:documentation>
      A group of controls allowed in all menus.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="control" type="CT_ControlCloneRegular">
      <xsd:annotation>
        <xsd:documentation>
          Control can enable, disable, or clone built-in controls.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="button" type="CT_ButtonRegular">
      <xsd:annotation>
        <xsd:documentation>
          Button control.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="checkBox" type="CT_CheckBox" >
      <xsd:annotation>
        <xsd:documentation>
          CheckBox control.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="gallery" type="CT_GalleryRegular">
      <xsd:annotation>
        <xsd:documentation>
          Gallery control.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="toggleButton" type="CT_ToggleButtonRegular">
      <xsd:annotation>
        <xsd:documentation>
          ToggleButton control.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="menuSeparator" type="CT_MenuSeparator">
      <xsd:annotation>
        <xsd:documentation>
          Control group separator.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
</xsd:group>

```

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:choice>
</xsd:group>
<xsd:group name="EG MenuOrSplitButtonRegular">
    <xsd:annotation>
        <xsd:documentation>
            Defines menu or splitButton controls.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
        <xsd:element name="splitButton" type="CT_SplitButtonRegular" >
            <xsd:annotation>
                <xsd:documentation>
                    SplitButton control.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="menu" type="CT_MenuRegular">
            <xsd:annotation>
                <xsd:documentation>
                    Menu.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="dynamicMenu" type="CT_DynamicMenuRegular">
            <xsd:annotation>
                <xsd:documentation>
                    DynamicMenu.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:choice>
</xsd:group>
<xsd:group name="EG MenuOrSplitButtonWithTitle">
    <xsd:annotation>
        <xsd:documentation>
            Menu or split button controls with title.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
        <xsd:element name="splitButton" type="CT_SplitButtonWithTitle" >
            <xsd:annotation>
                <xsd:documentation>
                    SplitButton control.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="menu" type="CT_MenuWithTitle">
            <xsd:annotation>
                <xsd:documentation>
                    Menu.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="dynamicMenu" type="CT_DynamicMenuRegular">
            <xsd:annotation>
                <xsd:documentation>
                    DynamicMenu.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="CT_OfficeMenu">
    <xsd:annotation>
        <xsd:documentation>

```

```

        The OfficeMenu control.
    </xsd:documentation>
</xsd:annotation>
</xsd:sequence>
<xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
        <xsd:group ref="EG_MenuControlsBase"/>
        <xsd:group ref="EG_MenuOrSplitButtonWithTitle"/>
    </xsd:choice>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_MenuRegular">
    <xsd:annotation>
        <xsd:documentation>
            A menu with a fixed-size button.
            Contains one or more controls or other menus.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ControlBase">
            <xsd:sequence>
                <xsd:choice minOccurs="0" maxOccurs="1000">
                    <xsd:group ref="EG_MenuControlsBase"/>

                    <xsd:group ref="EG_MenuOrSplitButtonRegular"/>
                </xsd:choice>
            </xsd:sequence>
            <xsd:attribute name="itemSize" type="ST_ItemSize" use="optional">
                <xsd:annotation>
                    <xsd:documentation>
                        The size of menu items.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attributeGroup ref="AG_Description"/>
            <xsd:attributeGroup ref="AG_IDAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_DynamicMenuRegular">
    <xsd:annotation>
        <xsd:documentation>
            A dynamicMenu with a fixed-size button.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ControlBase">
            <xsd:attributeGroup ref="AG_Description"/>
            <xsd:attributeGroup ref="AG_IDAttributes"/>
            <xsd:attributeGroup ref="AG_GetContentAttributes"/>
            <xsd:attributeGroup ref="AG_DynamicContentAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_MenuWithTitle">
    <xsd:annotation>
        <xsd:documentation>
            A menu with a fixed-size button.
            Contains one or more controls or other menus and has a 'title' attribute.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="CT_ControlBase">
            <xsd:sequence>
                <xsd:choice minOccurs="0" maxOccurs="1000">
                    <xsd:group ref="EG_MenuControlsBase"/>
                    <xsd:group ref="EG_MenuOrSplitButtonWithTitle"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

    <xsd:attributeGroup ref="AG_IDAttributes"/>
    <xsd:attribute name="itemSize" type="ST_ItemSize" use="optional">
      <xsd:annotation>
        <xsd:documentation>
          The size of menu items.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="AG_Title"/>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_Menu">
  <xsd:annotation>
    <xsd:documentation>
      A menu with a button that can be sized.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_MenuRegular">
      <xsd:attributeGroup ref="AG_SizeAttributes"/>
      <xsd:attribute name="itemSize" type="ST_ItemSize" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            The size of menu items. Large menu items show the 'description' attribute
inline.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_DynamicMenu">
  <xsd:annotation>
    <xsd:documentation>
      A dynamicMenu with a button that can be sized.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_DynamicMenuRegular">
      <xsd:attributeGroup ref="AG_SizeAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_SplitButtonBase">
  <xsd:annotation>
    <xsd:documentation>
      A splitButton-type control with a fixed-size.
      SplitButton contains a button or toggleButton and a menu.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_Control">
      <xsd:attributeGroup ref="AG_Enabled"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_SplitButtonRestricted">
  <xsd:annotation>
    <xsd:documentation>
      Attributes that do not apply to splitButton.
      They are inherited from the button inside of the splitButton.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="CT_SplitButtonBase">
      <xsd:attribute name="label" use="prohibited">

```

```

    <xsd:annotation>
      <xsd:documentation>
        Label does not apply to splitButton, set it on the button inside the
splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getLabel" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetLabel inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="screentip" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        Screentip inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getScreentip" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetScreentip inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="supertip" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        Supertip inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getSupertip" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetSuperTip inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="image" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        Image inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="imageMso" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        ImageMso inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetImage inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="showImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        ShowImage inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>

```



```

    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="getShowImage" use="prohibited">
    <xsd:annotation>
      <xsd:documentation>
        GetShowImage inherited from button inside splitButton.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_SplitButtonRegular">
  <xsd:annotation>
    <xsd:documentation>
      A splitButton with a fixed-size.
      SplitButton contains a button or toggleButton and a menu.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_SplitButtonRestricted">
      <xsd:sequence minOccurs="0">
        <xsd:choice minOccurs="0" >
          <xsd:element name="button" type="CT_VisibleButton">
            <xsd:annotation>
              <xsd:documentation>
                Button.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="toggleButton" type="CT_VisibleToggleButton" >
            <xsd:annotation>
              <xsd:documentation>
                ToggleButton.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:choice>
        <xsd:element name="menu" type="CT_MenuRegular" >
          <xsd:annotation>
            <xsd:documentation>
              Menu.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_SplitButtonWithTitle">
  <xsd:annotation>
    <xsd:documentation>
      A splitButton with a fixed-size, and title attribute.
      SplitButton contains a button or toggleButton and a menu.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_SplitButtonRestricted">
      <xsd:sequence minOccurs="0">
        <xsd:choice minOccurs="0" >
          <xsd:element name="button" type="CT_VisibleButton">
            <xsd:annotation>
              <xsd:documentation>
                Button.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="toggleButton"

```

```

type="CT_VisibleToggleButton" >
  <xsd:annotation>
    <xsd:documentation>
      ToggleButton.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:choice>
<xsd:element name="menu" type="CT_MenuWithTitle" >
  <xsd:annotation>
    <xsd:documentation>
      Menu.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_SplitButton">
  <xsd:annotation>
    <xsd:documentation>
      A splitButton that can be sized.
      SplitButton contains a button or toggleButton and a menu.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="CT_SplitButtonRegular">
      <xsd:attributeGroup ref="AG_SizeAttributes"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:group name="EG_Controls">
  <xsd:annotation>
    <xsd:documentation>
      Control types.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="control" type="CT_ControlClone">
      <xsd:annotation>
        <xsd:documentation>
          Control element can enable, disable or clone built-in controls.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="labelControl" type="CT_LabelControl" >
      <xsd:annotation>
        <xsd:documentation>
          LabelControl.
          Shows text and/or image.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="button" type="CT_Button" >
      <xsd:annotation>
        <xsd:documentation>
          Button control.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="toggleButton" type="CT_ToggleButton" >
      <xsd:annotation>
        <xsd:documentation>
          ToggleButton control, a button with on/off state.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>

```

```

    </xsd:annotation>
  </xsd:element>
  <xsd:element name="checkBox" type="CT_CheckBox" >
    <xsd:annotation>
      <xsd:documentation>
        CheckBox control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="editBox" type="CT_EditBox" >
    <xsd:annotation>
      <xsd:documentation>
        EditBox control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="comboBox" type="CT_Combobox" >
    <xsd:annotation>
      <xsd:documentation>
        ComboBox control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="dropDown" type="CT_DropDownRegular" >
    <xsd:annotation>
      <xsd:documentation>
        DropDown control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="gallery" type="CT_Gallery" >
    <xsd:annotation>
      <xsd:documentation>
        Gallery control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="menu" type="CT_Menu" >
    <xsd:annotation>
      <xsd:documentation>
        Menu control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="dynamicMenu" type="CT_DynamicMenu" >
    <xsd:annotation>
      <xsd:documentation>
        DynamicMenu control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="splitButton" type="CT_SplitButton" >
    <xsd:annotation>
      <xsd:documentation>
        SplitButton control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="box" type="CT_Box" >
    <xsd:annotation>
      <xsd:documentation>
        Box control.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="buttonGroup" type="CT_ButtonGroup" >
    <xsd:annotation>
      <xsd:documentation>

```

```

        ButtonGroup control.
    </xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:choice>
</xsd:group>

<xsd:annotation>
  <xsd:documentation>
    -----
    Containers
    -----
  </xsd:documentation>
</xsd:annotation>
<xsd:complexType name="CT_DialogLauncher">
  <xsd:annotation>
    <xsd:documentation>
      Opens a dialog related to its parent group.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="button" type="CT_ButtonRegular" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>
          Opens a dialog related to its parent group.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_Box">
  <xsd:annotation>
    <xsd:documentation>
      A box control, useful for grouping controls horizontally and vertically.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:group ref="EG_Controls" minOccurs="0" maxOccurs="1000" />
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
  <xsd:attribute name="boxStyle" type="ST_BoxStyle" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Flow of controls inside the box.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="CT_Separator">
  <xsd:annotation>
    <xsd:documentation>
      An in-Ribbon separator. Renders as a vertical bar.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
</xsd:complexType>
<xsd:complexType name="CT_MenuSeparator">
  <xsd:annotation>
    <xsd:documentation>
      A menuSeparator. Renders as a horizontal bar.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
  <xsd:attributeGroup ref="AG_Title"/>
</xsd:complexType>

```

```

<xsd:complexType name="CT_ButtonGroup">
  <xsd:annotation>
    <xsd:documentation>
      A ButtonGroup. A horizontal box with special visual appearance.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:element name="control" type="CT_ControlCloneRegular">
        <xsd:annotation>
          <xsd:documentation>
            Control element can enable, disable or clone built-in controls.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="button" type="CT_ButtonRegular" >
        <xsd:annotation>
          <xsd:documentation>
            Button.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="toggleButton" type="CT_ToggleButtonRegular" >
        <xsd:annotation>
          <xsd:documentation>
            ToggleButton control, a button with on/off state.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>

      <xsd:element name="gallery" type="CT_GalleryRegular">
        <xsd:annotation>
          <xsd:documentation>
            Gallery control.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="menu" type="CT_MenuRegular">
        <xsd:annotation>
          <xsd:documentation>
            Menu control.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="dynamicMenu" type="CT_DynamicMenuRegular">
        <xsd:annotation>
          <xsd:documentation>
            DynamicMenu control.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="splitButton" type="CT_SplitButtonRegular">
        <xsd:annotation>
          <xsd:documentation>
            SplitButton control.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attributeGroup ref="AG_IDCustom"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
</xsd:complexType>
<xsd:complexType name="CT_Group">
  <xsd:annotation>
    <xsd:documentation>
      A group.
    </xsd:documentation>
  </xsd:annotation>

```

```

</xsd:annotation>
<xsd:sequence>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:group ref="EG_Controls"/>
      <xsd:element name="separator" type="CT_Separator">
        <xsd:annotation>
          <xsd:documentation>
            Separator control.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
  <xsd:element name="dialogBoxLauncher" type="CT_DialogLauncher" minOccurs="0"
maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>
        DialogBoxLauncher. Opens a dialog related to its parent group.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
<xsd:attributeGroup ref="AG_IDAttributes"/>
<xsd:attributeGroup ref="AG_Label"/>
<xsd:attributeGroup ref="AG_Image"/>
<xsd:attributeGroup ref="AG_PositionAttributes"/>
<xsd:attributeGroup ref="AG_Screentip"/>
<xsd:attributeGroup ref="AG_Visible"/>
<xsd:attributeGroup ref="AG_Keytip"/>
</xsd:complexType>
<xsd:complexType name="CT_Tab">
  <xsd:annotation>
    <xsd:documentation>
      A tab that contains groups.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="100">
      <xsd:element name="group" type="CT_Group">
        <xsd:annotation>
          <xsd:documentation>
            Group. Contains controls.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attributeGroup ref="AG_IDAttributes"/>
  <xsd:attributeGroup ref="AG_Label"/>
  <xsd:attributeGroup ref="AG_PositionAttributes"/>
  <xsd:attributeGroup ref="AG_Visible"/>
  <xsd:attributeGroup ref="AG_Keytip"/>
</xsd:complexType>
<xsd:complexType name="CT_QatItems">
  <xsd:annotation>
    <xsd:documentation>
      Quick Access Toolbar items (shared or document-specific).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="1000">
      <xsd:element name="control" type="CT_ControlClone">
        <xsd:annotation>
          <xsd:documentation>
            Control element can enable, disable or clone built-in controls.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>

```

```

</xsd:element>
<xsd:element name="button" type="CT_ButtonRegular">
  <xsd:annotation>
    <xsd:documentation>
      Button control.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="separator" type="CT_Separator">
  <xsd:annotation>
    <xsd:documentation>
      Control group separator.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:choice>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_Qat">
  <xsd:annotation>
    <xsd:documentation>
      The Quick Access Toolbar.
      Two control collections: 1) Shared - applied to all windows and documents and 2)
      Document - attached to a document.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="sharedControls" type="CT_QatItems" minOccurs="0" >
      <xsd:annotation>
        <xsd:documentation>
          Set of controls shared between all windows or instances of the application.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="documentControls" type="CT_QatItems" minOccurs="0" >
      <xsd:annotation>
        <xsd:documentation>
          Set of controls attached to the current document.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_Tabs">
  <xsd:annotation>
    <xsd:documentation>
      A collection of tabs.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="tab" type="CT_Tab" minOccurs="1" maxOccurs="100">
      <xsd:annotation>
        <xsd:documentation>
          Tab.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_TabSet">
  <xsd:annotation>
    <xsd:documentation>
      A collection of contextual tab sets.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="tab" type="CT_Tab" minOccurs="0" maxOccurs="50">
      <xsd:annotation>

```

```

        <xsd:documentation>
            A Contextual Tab.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="idMso" type="ST_ID" use="required">
    <xsd:annotation>
        <xsd:documentation>
            The ID of a built-in control.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attributeGroup ref="AG_Visible"/>
</xsd:complexType>
<xsd:complexType name="CT_ContextualTabs">
    <xsd:annotation>
        <xsd:documentation>
            A collection of contextual tab sets.
        </xsd:documentation>
    </xsd:annotation>
</xsd:sequence>
    <xsd:element name="tabSet" type="CT_TabSet" minOccurs="1" maxOccurs="100">
        <xsd:annotation>
            <xsd:documentation>
                TabSet.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:annotation>
    <xsd:documentation>
        -----
        Root elements
        -----
    </xsd:documentation>
</xsd:annotation>

<xsd:complexType name="CT_Commands">
    <xsd:annotation>
        <xsd:documentation>
            A collection of Command elements.
        </xsd:documentation>
    </xsd:annotation>
</xsd:sequence>
    <xsd:element name="command" type="CT_Command" minOccurs="1" maxOccurs="5000">
        <xsd:annotation>
            <xsd:documentation>
                Attribute overrides for all controls with specified idMso.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_Ribbon">
    <xsd:annotation>
        <xsd:documentation>
            The Ribbon which contains the tabs, contextualTabs, officeMenu, and qat control
            types.
        </xsd:documentation>
    </xsd:annotation>
</xsd:all>
    <xsd:element name="officeMenu" type="CT_OfficeMenu" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
            <xsd:documentation>
                OfficeMenu.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:all>

```



```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="qat" type="CT_Qat" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>
            The Quick Access Toolbar.
            Two control collections: 1) Shared - applied to all windows and documents and 2)
Document - attached to a document.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="tabs" type="CT_Tabs" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>
            A collection of tabs.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="contextualTabs" type="CT_ContextualTabs" minOccurs="0"
maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>
            A collection of contextual tab sets.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:all>
<xsd:attribute name="startFromScratch" type="xsd:boolean" use="optional">
    <xsd:annotation>
        <xsd:documentation>
            Whether startFromScratch is enabled. It's a mode that hides almost all of the
standard UI.
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<xsd:complexType name="CT_MenuRoot">
    <xsd:annotation>
        <xsd:documentation>
            The root element of the customization XML returned by the getContent callback on
dynamicMenus.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="1000">
            <xsd:group ref="EG_MenuControlsBase"/>
            <xsd:group ref="EG_MenuOrSplitButtonRegular"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="AG_Title"/>
    <xsd:attribute name="itemSize" type="ST_ItemSize" use="optional">
        <xsd:annotation>
            <xsd:documentation>
                The size of menu items.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:element name="menu" type="CT_MenuRoot">
    <xsd:annotation>
        <xsd:documentation>
            The root element of the customization XML used in dynamicMenu's getContent callback.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="CT_CustomUI">

```

```

    <xsd:annotation>
      <xsd:documentation>
        The root element of the customization file which is used to create or modify the
        Ribbon and other UI elements.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:sequence>
  <xsd:sequence>
    <xsd:element name="commands" type="CT_Commands" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>
          Command overrides.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="ribbon" type="CT_Ribbon" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>
          Ribbon.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="onLoad" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Callback invoked when custom UI is loaded.
        IRibbonUI object is passed as a parameter. This object exposes Invalidate and
        InvalidateControl.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="loadImage" type="ST_Delegate" use="optional">
    <xsd:annotation>
      <xsd:documentation>
        A callback to load all images. If specified, all controls with the image attribute
        set will call this callback with the attribute value passed as a string.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:element name="customUI" type="CT_CustomUI">
  <xsd:annotation>
    <xsd:documentation>
      The root element used to create or modify the Ribbon and other UI components.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:schema>

```

5 Appendix C: Product Behavior

- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Microsoft Office 2016
- Microsoft Office 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.

6 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
4.1 http://schemas.microsoft.com/office/2006/01/customui Schema	Added full XML schema.	Minor
5 Appendix C: Product Behavior	Updated list of supported products.	major

7 Index

C

[Change tracking](#) 554
Custom UI
 [parts](#) 8
Custom UI control id table
 [imageMso table](#) 417

E

[Elements](#) 10
 [box \(box grouping container\)](#) 10
 [button \(button inside of a split button\)](#) 33
 [button \(button\)](#) 14
 [button \(unsized button\)](#) 24
 [buttonGroup \(button grouping container\)](#) 41
 [checkBox \(check box\)](#) 45
 [comboBox \(combo box\)](#) 54
 [command \(repurposed command\)](#) 66
 [commands \(list of repurposed commands\)](#) 67
 [contextualTabs \(list of contextual tab sets\)](#) 68
 [control \(control clone\)](#) 76
 [control \(Quick Access Toolbar control clone\)](#) 86
 [control \(unsized control clone\)](#) 68
 [customUI \(custom UI document root\)](#) 95
 [dialogBoxLauncher \(dialog box launcher\)](#) 95
 [documentControls \(list of document-specific Quick Access Toolbar controls\)](#) 96
 [dropDown \(drop-down control\)](#) 97
 [dynamicMenu \(dynamic menu\)](#) 118
 [dynamicMenu \(unsized dynamic menu\)](#) 109
 [editBox \(edit box\)](#) 129
 [gallery \(gallery\)](#) 138
 [gallery \(unsized gallery\)](#) 154
 [group \(group\)](#) 168
 [item \(selection item\)](#) 176
 [labelControl \(text label\)](#) 178
 [menu \(dynamic menu root XML element\)](#) 217
 [menu \(menu with title\)](#) 196
 [menu \(menu\)](#) 206
 [menu \(unsized menu\)](#) 187
 [menuSeparator \(menu separator\)](#) 219
 [officeMenu \(Office menu\)](#) 222
 [qat \(Quick Access Toolbar\)](#) 223
 [ribbon \(ribbon\)](#) 224
 [separator \(separator\)](#) 225
 [sharedControls \(list of shared Quick Access Toolbar controls\)](#) 228
 [splitButton \(split button with title\)](#) 237
 [splitButton \(split button\)](#) 246
 [splitButton \(unsized split button\)](#) 229
 [tab \(tab\)](#) 255
 [tabs \(list of tabs\)](#) 259
 [tabSet \(contextual tab set\)](#) 260
 [toggleButton \(toggle button inside of a split button\)](#) 280
 [toggleButton \(toggle button\)](#) 270
 [toggleButton \(unsized toggle button\)](#) 261

G

[Glossary](#) 6

I

idMso tables
 [Excel 2007](#) 356
 [PowerPoint 2007](#) 391
 [Word 2007](#) 302
[Informative references](#) 7
[Introduction](#) 6

N

[Normative references](#) 7

P

Parts
 [additional part types](#) 8
 [quick access toolbar customizations part](#) 8
 [ribbon extensibility part](#) 9
[Product behavior](#) 553

R

References
 [informative](#) 7
 [normative](#) 7

S

[Simple types](#) 289
 [ST_BoxStyle \(box style\)](#) 289
 [ST_Delegate \(callback function name\)](#) 290
 [ST_GalleryItemWidthHeight \(gallery item width or height\)](#) 292
 [ST_GalleryRowColumnCount \(gallery row or column count\)](#) 293
 [ST_ID \(control identifier\)](#) 293
 [ST_ItemSize \(menu item size\)](#) 294
 [ST_Keytip \(key tip\)](#) 295
 [ST_LongString \(long string\)](#) 295
 [ST_QID \(qualified control identifier\)](#) 296
 [ST_Size \(control size\)](#) 298
 [ST_String \(short string\)](#) 298
 [ST_StringLength \(string length\)](#) 299
 [ST_UniqueID \(custom control identifier\)](#) 300
 [ST Uri \(image relationship identifier\)](#) 300

T

Tables
 [idMso table – Excel 2007](#) 356
 [idMso table – PowerPoint 2007](#) 391
 [idMso table – Word 2007](#) 302
 [imageMso table](#) 417
[Tracking changes](#) 554