Adding Icons to Menu Commands

本文内容

Adding an Icon to a Command

See Also

Commands can appear on both menus and toolbars. On toolbars, it is common for a command to be displayed with just an icon (to save space) while on menus a command typically appears with both an icon and text.

Icons are 16 pixels wide by 16 pixels high and can be either 8-bit color depth (256 colors) or 32-bit color depth (true color). 32-bit color icons are preferred. Icons are typically arranged in a single horizontal row in a single bitmap, although multiple bitmaps are allowed. This bitmap is declared in the .vsct file along with the individual icons available in the bitmap. See the reference for the [Bitmaps Element](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/bitmaps-element) for more details.

Adding an Icon to a Command

The following procedure assumes that you have an existing VSPackage project with a menu command. To find out how to do this, see [Creating an Extension with a Menu Command](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/creating-an-extension-with-a-menu-command).

**1.** Create a bitmap with a color depth of 32-bits. An icon is always 16 x 16 so this bitmap must be 16 pixels high and a multiple of 16 pixels wide.

Each icon is placed on the bitmap next to each other in a single row. Use the alpha channel to indicate places of transparency in each icon.

If you use an 8-bit color depth, use magenta, RGB(255,0,255), as the transparency. However, 32-bit color icons are preferred.

**2.** Copy the icon file to the Resources directory in your VSPackage project. In the Solution Explorer, add the icon to the project. (Select Resources, and on the context menu click Add, then Existing Item, and select your icon file.)

**3.** Open the .vsct file in the editor.

**4.** Add a GuidSymbol element with a name of **testIcon**. Create a GUID (**Tools / Create GUID**, then select **Registry Format** and click Copy) and paste it into the value attribute. The result should look like this:

<!-- Create your own GUID -->

<GuidSymbol name="testIcon" value="{00000000-0000-0000-0000-0000}">

**5.** Add an <IDSymbol> for the icon. The name attribute is the icon's ID, and the value indicates its position on the strip, if any. If there is just one icon, add 1. The result should look like this:

<!-- Create your own GUID -->

<GuidSymbol name="testIcon" value="{00000000-0000-0000-0000-0000}">

<IDSymbol name="testIcon1" value="1" />

</GuidSymbol>

**6.** Create a <Bitmap> in the <Bitmaps> section of the .vsct file to represent the bitmap containing the icons.

• Set the guid value to the name of the <GuidSymbol> element you created in the previous step.

• Set the href value to the relative path of the bitmap file (in this case **Resources\**.

• Set the usedList value to the IDSymbol you created earlier. This attribute specifies a comma-delimited list of the icons to be used in the VSPackage. Icons not on the list are excluded form compilation.

The Bitmap block should look like this:

<Bitmap guid="testIcon" href="Resources\<icon file name>" usedList="testIcon1"/>

**7.** In the existing <Button> element, set the Icon element to the GUIDSymbol and IDSymbol values you created earlier. Here's an example of a Button element with those values:

<Button guid="guidAddIconCmdSet" id="cmdidMyCommand" priority="0x0100" type="Button">

<Parent guid="guidAddIconCmdSet" id="MyMenuGroup" />

<Icon guid="testIcon" id="testIcon1" />

<Strings>

<ButtonText>My Command name</ButtonText>

</Strings>

</Button>

**8.** Test your icon. Build the project and start debugging. In the experimental instance, find the command. It should show the icon you added.

See Also

[Extending Menus and Commands](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/extending-menus-and-commands)

[VSCT XML Schema Reference](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/vsct-xml-schema-reference)