API Ribbon Layout Guidelines

1 Introduction

The following are aspects of the ribbon UI that can be modified by individual add-in developers. These guidelines must be followed to make your application's user interface (UI) compliant with standards used by Autodesk.

1.1 Terminology

Several words are used to signify the requirements of the standards. These words are capitalized. This section defines how these special words should be interpreted. The interpretation has been copied from **Internet Engineering Task Force RFC 2119**. Some minor modifications were added.

- **MUST**: This word or the term "SHALL", mean that the item is an absolute requirement.
- **MUST NOT:** This phrase, or the phrase "SHALL NOT", means that the item is an absolute prohibition.
- **SHOULD:** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore the item, but the full implications must be understood and carefully weighed before choosing a different course.
- **SHOULD NOT:** This phrase, or the phrase "NOT RECOMMENDED", mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- **MAY:** This word, or the adjective "OPTIONAL", means that the item is truly optional. One product team may choose to include the item because a particular type of user requires it or because the product team feels that it enhances the product while another product team may omit the same item.

2   Definitions

2.1   Ribbon

The horizontal tabbed user interface across the top of the application frame in Revit 2010 products.

2.2   Ribbon Tab

The ribbon is separated into tabs. The Add-Ins ribbon tab, which only appears when at least one add-in is installed, is available for third party developers to add a panel.

2.3   Panel

A ribbon tab is separated into horizontal groupings of commands. An Add-In panel represents the commands available for a third party developer's application. The Add-In panel is equivalent to the toolbar in Revit 2009.

2.4   Button

The button is the mechanism for launching a command. They can either be large (see Measure in Figure 1) or small (see Report Coordinates in Figure 1.) Both large and small buttons can either be a simple push button or a drop-down button (see below.)
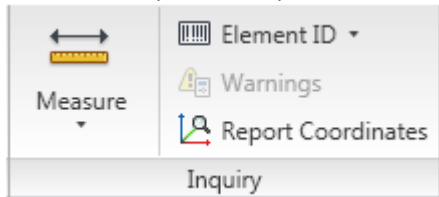

Figure 1 - Inquiry panel in the 2010 Revit products

2.5   Menu button

The default first panel on the Add-Ins tab is the External Tools panel that contains one button titled "External Tools.". The External Tools menu-button is equivalent to the Tools > External Tools menu in Revit 2009. Any External Command registered in Revit.ini under [ExternalCommands] will appear in this menu button.
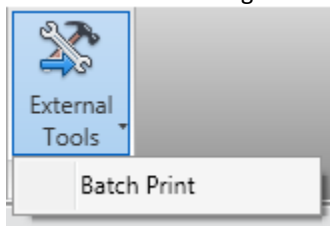

Figure 2 - External Tools menu-button on Add-Ins tab

2.6   Drop-down button

A drop-down button expands to show two or more commands in a drop-down menu. Each sub-command can have its own large icon (see Figure 4 below.)

2.7   Vertical Separator

A vertical separator is a thin vertical line that can be added between controls on a panel (see to the right of Measure button in Figure 3 below.)

2.8   Tooltip

A tooltip is a small panel that appears when the user hovers the mouse pointer over a ribbon button. Tooltips provide a brief explanation of the command's expected behavior.

3 Layout Guidelines

3.1 Number of Panels per Tab

Each API application SHOULD add only one panel to the Add-Ins tab.

3.2 Panel Layout

The following guidelines define the proper way to lay out a panel on the Add-ins tab. The following panel from Revit 2010 provides an example to follow.
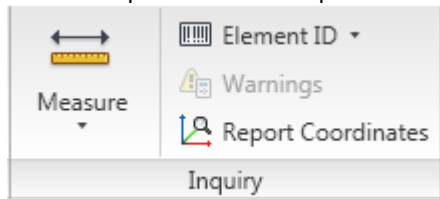

Figure 3 - Inquiry panel in the 2010 Revit products

3.2.1 General layout

A panel SHOULD have a large button as the left-most control. This button SHOULD be the most commonly accessed command in the application. The left-most button icon will represent the entire panel when it collapses (see 3.3 below.)This button MAY be the only button in the group, or this button MAY be followed by a large button and/or a small button stack.

Panels SHOULD NOT exceed three columns. If more controls are necessary, use a drop-down button (see 3.2.3 below.)

3.2.2 Small button stack:

- The stack MUST have at least two buttons and MUST NOT exceed three.
- The order of the small buttons SHOULD follow most frequent on bottom to least frequent on top. This is because the more frequently accessed command should be closer to the modeling window.

3.2.3 Drop-down button:

- The top label SHOULD sufficiently describe the contents of the drop-down list. For example, in Figure 4, Roof, Curtain System, Wall and Floor are all ways of modeling by face.)
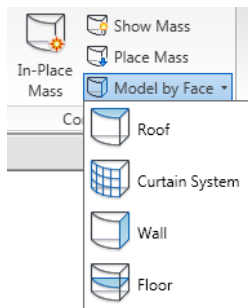- Every item in the list SHOULD contain a large icon.

Figure 4 - Model by Face drop-down button in Revit Architecture 2010.

3.2.4   Vertical separator

A vertical separator MAY be added between a control or sets of controls to create distinct groupings of commands within a panel. For example, in Figure 3, the Measure button deals with measuring elements while the items in the three button stack deal with querying data from the model. A panel SHOULD have no more than two separators.

3.2.5   Icons

For proper icon design, see the Autodesk Icon Guidelines.pdf elsewhere in the SDK.

3.3   Panel Resizing and Collapsing

By default, panels will be placed left to right in descending order left to right based on the order in which they were installed by the customer. Once the width of the combined panels exceeds the width of the current window, the panels will start to resize starting from the right in the following order:
1. Panels with large buttons:
     a. Small buttons lose their labels, and then:
     b. The panel collapses to a single large button (the icon representing the panel will be the first icon on the left.)
2. Panels with ONLY small button stack(s)
     a. Small buttons lose their labels and the panel label gets truncated to four characters and an ellipsis (three periods in a row.)
     b. If a small button stack is the left-most control in a panel, then the top button must have a large icon associated with it. This icon will represent the panel when collapsed.

Note: Panel resizing and collapsing is handled automatically by the ribbon component.

3.4   Button Labels

These guidelines are for English language only.

3.4.1   MUST not have any punctuation (except hyphen, ampersand or forward slash, see below)

3.4.2   MUST be no longer than three words

3.4.3   MUST be no longer than 36 characters

3.4.4   MUST be Title Case; e.g. Show Mass

3.4.5   The ampersand '&' MUST be used instead of 'and'. A space should appear before and after the ampersand.

3.4.6   The forward slash '/' MUST be used instead of 'or'. No spaces should appear before and after the slash.

3.4.7   Only large buttons MAY have two line labels but MUST NOT have more than two lines. Labels for all other widgets MUST fit on a single line.

3.4.8   Button labels MUST NOT contain ellipses (…).

3.4.9   Every word MUST be in capital case except articles ("a," "an," and "the"), coordinating conjunctions (for example, "and," "or," "but," "so," "yet," "with," and "nor"), and prepositions with fewer than four letters (like "in"). The first and last words are always capitalized, regardless of what they are.

3.5   Panel Labels

These guidelines are English-only. All rules from the Button Labels section apply to Panel Labels in addition to the following:

3.5.1   The name of the panel SHOULD be specific. Vague, non-descriptive and unspecific terms used to describe panel content will reduce the label's usefulness.

3.5.2   Applications MUST NOT use panel names that use the abbreviations 'misc.' or 'etc.'

3.5.3   Panel labels SHOULD NOT include the term 'add-ins' since it is redundant with the tab label.

3.5.4   Panel labels MAY include the name of the third party product or provider.

3.6   Tooltips

The following are guidelines for writing tooltip text. Write concisely. There is limited space to work with.

3.6.1   Make every word count. These guidelines are particularly important for localizing tooltip text to other languages.

3.6.2   Use simple sentences. The "Verb-Object-Adverb" format is recommended.

3.6.3   Use only one space between sentences.

3.6.4   Avoid repetitive text. The content in the tooltip should be unique and add value.

3.6.5   Don't include lengthy step-by-step procedures in tooltips. These belong in Help.

3.6.6   Use terminology consistently.

3.6.7   Don't use gerunds (verb forms used as nouns) because they can be confused with participles (verb forms used as adjectives). In the example, "Drawing controls", drawing could be used as a verb or a noun. A better example is "Controls for drawing."

3.6.8   Focus on the quality and understandability of the tooltip. Is the description clear? Is it helpful?

3.6.9   Use strong and specific verbs that describe a specific action (such as "tile") rather than weak verbs (such as "use to…").

3.6.10   Write in the active voice (for example, "Moves objects between model space and paper space").

3.6.11   Use the descriptive style instead of the imperative style ("Opens an existing drawing file" vs. "Open an existing drawing file").

3.6.12   Unless it's a system variable or command, do not use bold. Although bold is supported in Asian languages, it is strongly recommended to avoid using bold and italics, because of readability and stylistic issues.

3.6.13   Make the tooltip description easily recognizable by using the third person singular (for example – "Specifies the current color" instead of "Specify the current color".)

Don't use slang, jargon, or hard to understand acronyms.

3.6.14   Make sure that your use of conjunctions doesn't introduce ambiguities in relationships. For example, instead of saying "replace and tighten the hinges", it would be better to split the conjunction up into two simple (and redundant) sentences – "Replace the hinges. Then tighten the hinges".

3.6.15   Be careful with "helping" verbs. Examples of helping verbs include *shall, may, would have, should have, might have, and can.* For example, *can* and *may* could be translated as "capability" and "possibility" respectively.

3.6.16   Watch for invisible plurals such as "object and attribute settings". Does this mean "the settings for one object and one attribute" or "the settings for many objects and many attributes"?

3.6.17 Be cautious about words that can be either nouns or verbs. Use articles or rewrite phrases like "Model Display" where model can be a noun or a verb in our software. Another example is "empty file". It can mean "to empty a file" or "a file with no content".

3.6.18 Be careful using metaphors. Metaphors can be subtle. Metaphors are often discussed in the context of icons that are not culturally appropriate or understood across cultures, but text metaphors (such as "places the computer in a hibernating state") can also be an issue. Instead, you might say "places the computer in a low-power state".

3.6.19 Avoid abbreviations. For example, the word "Number" has many common abbreviations: No., Nbr, Num, Numb. It is best to spell out terms.

Bad Example:
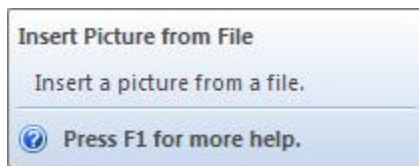
Insert Picture from File Insert a picture from a file. ⓘ Press F1 for more help.	In this example, the tooltip content repeats the tooltip title verbatim and does not add value to the tooltip. Additionally, there are translation issues with this example. If the translator can't identify whether this string is a name/title or a descriptive sentence, it will be difficult for them to decide on the translation style.

Good Example:
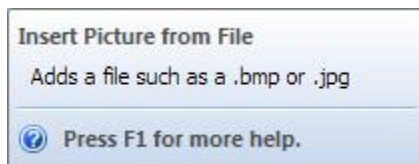
Insert Picture from File Adds a file such as a .bmp or .jpg ⓘ Press F1 for more help.	An example of a more useful descriptive sentence might be "Adds a file such as a .bmp or .jpg". This provides more detailed information and gives the user more insight into the feature.