

Assigning Keyboard Shortcuts to GUI Components

This document describes how to assign keyboard shortcuts to button and menu components in TIBCO General Interface™ applications. Keyboard shortcuts allow end users to interact with these components in applications without using a mouse.

Version 2.0. May 7, 2007

Scope: TIBCO General Interface 3.4



<http://www.tibco.com>

Global Headquarters

3303 Hillview Avenue
Palo Alto, CA 94304
Tel: +1 650-846-1000
Toll Free: 1 800-420-8450
Fax: +1 650-846-1005

© 2006-2007, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, The Power of Now, and TIBCO Software are trademarks or registered trademarks of TIBCO Software Inc. in the United States and/or other countries. All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

Table of Contents

Overview	3
Allowed Keys for Keyboard Shortcuts	3
Examples	3
Keyboard Shortcut Collisions	4
TIBCO General Interface Keyboard Shortcuts	4
Assigning Keyboard Shortcuts to Buttons	4
Assigning Keyboard Shortcuts to Menus	5
Keyboard Shortcuts Registration and Deployment.....	5

Overview

This document describes how to add keyboard shortcuts to your General Interface™ applications.

A keyboard shortcut allows a user to interact with menu and button components in an application without using the mouse. For example, a user might enter a keyboard shortcut to activate a menu command. Because some users prefer or require keyboard accessibility, it's important to add keyboard shortcuts to your application. This makes your application more flexible and user-friendly.

The following TIBCO General Interface components support keyboard shortcuts:

- `jx3.gui.Button`
- `jx3.gui.ImageButton`
- `jx3.gui.ToolbarButton`
- `jx3.gui.Menu`

This document assumes that you are familiar with adding GUI components in TIBCO General Interface. For more information, see *TIBCO General Interface Getting Started* in the online documentation.

Allowed Keys for Keyboard Shortcuts

When you assign keyboard shortcuts to a menu or a button, you use the following String format:

```
[ctrl+][alt+][shift+]n
```

The modifiers (`ctrl`, `alt`, `shift`) are optional and 0 to 3 modifiers can be used. The value `n` is required as the last value and can be any of the following keys:

- Alpha numeric characters: a-z, 0-9
- Punctuation keys in the string: ; (semi-colon) , (comma) . (period) / (forward slash) \ (back slash) ' (single quote) [] (open and close brackets) – (hyphen) = (equals sign) ` (back tick)
- Functions keys: F1-F15
- Special keys: enter, esc, tab, del, space, backspace, up, down, left, right, insert, home, end, pgup, pgdn.

Important: Lowercase is required when typing shortcut values in JavaScript code or in properties. For example, `ctrl+alt+b`.

Examples

```
ctrl+shift+enter  
ctrl+6  
ctrl+alt+s  
ctrl+,
```

Important: Although keyboard shortcuts can be reused in your application, don't reuse them in the same scope. For example, don't reuse the same keyboard shortcuts in a dialog.

Caution: Some shortcuts could be problematic. For example, if the shortcut for a button is `b` or `shift+b` and the end user is typing `b` or a capital `B` (`shift+b`) in a text field, the button will be activated instead. Choose your keyboard shortcuts carefully.

Keyboard Shortcut Collisions

Be aware that some keyboard shortcuts might already be in use. If you attempt to use these shortcuts, collisions could occur. For example, General Interface™ Builder uses Tab and Windows uses Alt+Tab for application switching. To avoid collisions with keyboard shortcuts already in use, be sure to thoroughly test your shortcuts in your running application.

There are four possible areas of collision for keyboard shortcuts:

- Operating system
- Browser
- General Interface Builder
- General Interface™ Framework

For information on the operating system and browser shortcuts, see the respective documentation.

TIBCO General Interface Keyboard Shortcuts

The following General Interface keyboard shortcuts are used by General Interface:

- Menu shortcuts used by General Interface Builder. For a list of menu shortcuts, see **Tools > IDE Settings > IDE Hot Keys** in General Interface Builder. Note that these menu shortcuts are customizable.
- Navigation shortcuts used by General Interface Builder and General Interface applications — tab, enter, space

If you want to use General Interface Builder shortcuts in your application, such as ctrl+n, your shortcut will only work in General Interface Builder if the focus is in the Live Component view of the shortcut component. Otherwise, the General Interface Builder shortcut executes. The navigation shortcuts should not be used.

Assigning Keyboard Shortcuts to Buttons

Keyboard shortcuts can be assigned to any of the following button components by modifying the Key Binding property:

- Button
- ImageButton
- ToolbarButton

To assign a keyboard shortcut to a button, do the following:

1. Add a **Button** component to your application from the **System > Form Elements** folder of the Component Library palette. Note that **ToolbarButton** is located in the System > Menus and Toolbars folder of the Component Libraries palette.
2. Select the **Button** component in the Component Hierarchy palette or the Live Component view.
3. Modify the **Key Binding** property as follows:
 - a. Open the **Property Editor palette** and click in the **Key Binding** value field.
 - b. Type in a keyboard shortcut, such as `ctrl+shift+enter`. It's recommend that you test the shortcut before assigning it to be sure it's not already used.

To test the shortcut,

1. Click anywhere in the Live Component view to give focus to your application.
Note: Don't click the button as this will activate it.
2. Press `ctrl+shift+enter` on the keyboard to test the button functionality.

Assigning Keyboard Shortcuts to Menus

To assign a keyboard shortcut to a menu, you need to add or modify the CDF @jsxkeycode attribute in the CDF document associated with the control.

To assign a keyboard shortcut to a menu command, do the following:

1. Add a **Menu** component to your application from the **System > Menus and Toolbars** folder of the Component Libraries palette.
2. Edit the CDF document associated with the control and add the @jsxkeycode attribute to each menu command—New, Open, and Save. There are several ways to do this:
 - Edit the CDF document on disk and enter the XML URL in the Properties Editor.
 - Edit the CDF cache document and enter the XML Cache ID in the Properties Editor (The cache document isn't persisted).
 - Edit the XML (String) property in the Properties Editor.

In this example, you'll edit the XML (String) property.

- a. Select the **Menu** component in the Component Hierarchy palette or the Live Component view.
- b. Open the **Property Editor palette** and click in the XML (String) value field.
- c. Add @jsxkeycode attributes to each menu command. Typically, the jsxexecute attribute would execute a function. In this example, an alert is used for the jsxexecute attribute for testing purposes to demonstrate that the keyboard shortcut is working properly.

For example, your XML (String) value might look similar to this:

```
<data jsxid="jsxroot">
  <record jsxid="1" jsxttext="New" jsxkeycode="alt+shift+n" jsxexecute="alert('New menu is working');"/>
  <record jsxid="2" jsxttext="Open" jsxkeycode="shift+o" jsxexecute="alert('Open menu is working');"/>
  <record jsxid="3" jsxttext="Save" jsxkeycode="shift+s" jsxexecute="alert('Save menu is working');"/>
</data>
```

Note: It's recommended that you test the shortcut before assigning it to be sure it's not already used.

To test the shortcut,

1. Select the **Menu** component in the Component Hierarchy palette and click the Re-Fetch Data and Repaint button so that the CDF document in memory reflects the changes made to the XML (String) property.
2. Click anywhere in the Live Component view to give focus to your application.
3. Use the keyboard shortcut for each menu command to test the functionality and to display the alert:
 - a. Press alt+shift+n for the New command.
 - b. Press shift+o for the Open command.
 - c. Press shift+s for the Save command.

Keyboard Shortcuts Registration and Deployment

When keyboard shortcuts are assigned, they are registered with the closest ancestor DOM node that is a jsx3.gui.Window, jsx3.gui.Dialog, or root block of a server (JSXROOT).

Therefore, shortcuts are global as there are no "protected" spaces. It is an error for two or more controls within the same scope to register the same shortcut.

There are two choices for deploying applications with keyboard shortcuts:

- Deploy with Body Hotkeys set to Yes
- Deploy with Body Hotkeys set to No

The first deployment choice, Body Hotkeys set to Yes, is typically used for full-window applications. In this deployment scenario, if the keyboard shortcut event bubbles up to the HTML <body> element, it is then sent to the application. So even if the focus is in the browser window and not in the General Interface application, the application receives the shortcut and executes it.

The second choice, Body Hotkeys set to No, is recommend for General Interface applications deployed in a page with other General Interface applications. In this case, the keyboard shortcuts only function if the focus is within the General Interface portlet application. This deployment prevents your application from responding to shortcuts initiated in other General Interface portlet applications on the page.

To set this option, choose **Project > Project Settings > Deployment**. The default setting is No.