2.3.1 Tab Panel – alternative implementation

Description: NO CHANGE

A tabbed interface component is a container for resources associated with a tab. It is a set of layered pages where only one page is displayed at a time. The general look is similar to a file folder with a "tab" that contains the title of the folder. The tabs are arranged along one of the edges of the contents but most commonly are found at the top of the page. The user navigates and makes the contents of each page visible by interacting with the title "tab" of the page. Sometimes referred to as a tab container or tab panel. Terms for understanding Tab Panels include:

tabbed interface component

a set of tabs and associated tab panels

tab panel

contents area that is associated with a tab

tab

the label/title area of the tab panel. This is where you click to activate a tab panel

tablist

the set of tabs

When the user activates a tab, the contents of the corresponding tab panel is made visible. The tab is considered "active". The tab remains active until another tab is activated. The active tab is placed into the tab order. Only the active tab should be in the tab order. A default tab is specified that is active when the tabbed interface component is initialized. A collection of tabs and their associated tab panels is a complex widget, because it performs show/hide actions as well as moving the user's point of regard around within the content.

Keyboard Interaction (alternative implementation):

This implementation does not activate the tab when it receives focus. In this scenario, the user must explicitly press the enter key to activate the tab with focus. This implementation allows the user to quickly navigate through the tab list without having to wait for each panel to load. The user navigates to the desired tab and presses the enter of space key to activate it.

- Tab - only the active tab is in the tab order. The user reaches the tabbed panel component by pressing the tab key until the active tab title receives focus.

- *Enter/Space key* – ***With focus on a tab, pressing the enter or space key will activate the tab and load the tab panel contents. This activation method is used to avoid activating and loading each tab panel as the user arrows through the tab list to reach the desired tab.***
- Left Arrow - with focus on a tab, pressing the left arrow will move focus to the previous tab in the tab list ~~*and activate that tab*~~. Pressing the left arrow when the focus is on the first tab in the tab list will move focus to ~~*and activate*~~ the last tab in the list.
- Right Arrow - with focus on a tab, pressing the right arrow will move focus to the next tab in the tab ~~*list and activate that tab*~~. Pressing the right arrow when the focus is on the last tab in the tab list will move focus to ~~*and activate*~~ the first tab in the list.
- Up arrow - behaves the same as left arrow in order to support vertical tabs
- Down arrow - behaves the same as right arrow in order to support vertical tabs
- Control+Up Arrow - with focus anywhere within the tab panel, pressing Control+Up Arrow will move focus to the tab for that panel. This is not standard behavior - is this something we want to implement? Is it necessary if we provide a mechanism to change the active tab? Similar to Control+PageUp/Control+PageDown in Firefox to switch tabs?
- Alt+Delete - When deletion is allowed, with focus anywhere within the tab panel, pressing Alt+Delete will delete the current tab and tab panel from the tabbed interface control. If additional tabs remain in the tabbed interface, focus goes to the next tab in the tab list. An alternative to providing a keystroke to close a tab is to provide a context menu that is associated with the tab title. When focus is on the tab, pressing Shift+F10 or pressing the right mouse button will open a context menu with the close choice
- Control+PageUp - When focus is inside of a tab panel, pressing Control+PageUp moves focus to the tab of the previous tab in the tab list ~~*and activates that tab*~~. When focus is in the first tab panel in the tab list, pressing Control+PageUp will move focus to the last tab in the tab list ~~*and activate that tab*~~.
- Control+PageDown When focus is inside of a tab panel, pressing Control+PageDown moves focus to the tab of the next tab in the tab list ~~*and activates that tab*~~. When focus is in the last tab panel in the tab list, pressing Control+PageUpwill move focus to the first tab in the tab list ~~*and activate that tab*~~.

Regarding Control+PageUp/Control+PageDown. This is currently implemented in Firefox to move between browser tabs. Firefox also supports Control+Tab and Control+Shift+Tab to move between tabs. Internet Explorer 7 also uses Control+Tab and Control+Shift+Tab. There may be advantages to using Control+PageUp/Control+PageDown as the keys to change tabs since it is a recognizable keystroke to at least Firefox users and is also supported by the Windows operating system to move between panels in a tabbed dialog. The problem is that if the user is within a tabbed interface control on a Web page, they can not easily switch browser tabs without first moving focus outside of the tabbed interface control. This may be acceptable. The other issue is if the entire Web page is a tabbed interface control - in that case the user could not ever switch browser tabs unless the control on the Web page ignored the Control+PageUp/Control+PageDown keypress (and thus letting the browser access it) when the first or last tab was reached.