

[0124] The following is yet another sample command:

```
[0125] Math=
      CMD:c:\winnt\system32\calc.exe;Favorites=http://
      www.msn.com;
```

[0126] By using of a theme called “Retro”, and the user then click the button with the command above, the Zenu™ UDI will not use either command.

[0127] Instead, the Zenu™ UDI will look for the default command for the button in the template resource file.

[0128] 5. Click and Drag

[0129] Users can click and drag shortcuts from the Windows desktop or Windows Explorer to a Zenu™ button of the present invention. This will cause the button to have the same action as the shortcut. If a file that is not a shortcut is dragged from Windows Explorer to a Zenu™ button, the Zenu™ UDI will make the button a shortcut pointing to the file that was dragged. For instance, if the user drags a Microsoft Word or notepad document onto a Zenu™ button, clicking that Zenu™ button will now open the document that was dragged onto the button. This overrides the default action of the button defined in the template or theme file as well as user-defined commands.

[0130] 6. Hot Key

[0131] A “Hot Key” can be assigned by the user to show/hide the Zenu™ UDI. For example, a specific, default set of key strokes, say Alt+F10, can be used. The user can readily change the default Hot Key, by right clicking anywhere on the UDI, and accessing “customize” and then the “Hot Key” feature. Next, all the user needs to do is press the desired key combination for the Hit Key, and it will be recorded in the text box of the Hot Key window. When the user is finished selecting the desired Hot Key, the user simply clicks “OK”. This will change the Hot Key, and store it so that the next time Zenu™ UDI is executed, it will use the same Hot Key combination to show/hide.

[0132] D. Sizing of the Zenu™ UDI

[0133] The Zenu™ customizer of the present invention also permits the user to change the size of the Zenu™ UDI (i.e., the space occupied on the screen by the Zenu™ UDI). Sizing can be an integral component of the tool, permitting the user to scale (stretch or shrink) the Zenu™ UDI to match the desired size. Sizing could be arbitrary, as in permitting the user click on an edge or “handle” and changed the size of the Zenu™ UDI. In a preferred embodiment, however, the user would be presented with a finite number of size option, say three sizes: smaller, normal, and larger. This allows exact scaling of the Zenu™ UDI to eliminate distortion and to maintain its aspect ratio. In another embodiment, up to ten different sizes are available including “full screen”, which becomes an attractive option when the Zenu™ UDI acts as the container for chat, video, browsing, mail, and the like.

[0134] The template and the theme can be designed to include knowledge of the multiple sizes available. Alternatively, all that is required is the “Normal” set of templates and theme components. As the user selects to change the size, the template/theme combination is scanned for prior-knowledge of possible sizes. The customizer can presents user with the additional choice(s). Once a new size is chosen, imagery designed specifically for the different size

is used. In the event that an image is not provided or available at the different size, the normal image is scaled to match the destination, as would become apparent to a person skilled in the art of developing window-based applications. Sizing of the Zenu™ UDI provides greater flexibility when implemented with the templates and themes. The system that allows the designers to choose whether they desired to re-use a simple graphic at multiple resolutions, or to duplicate an image for different resolutions increasing or decreasing the amount of detail included in the image. This is similar to conventional icon on the desktop; an icon (.ico) can contain up to four different images, two each in black and white and color at 16x16 and 32x32 pixels.

[0135] E. Exemplary Architecture Operation

[0136] FIGS. 12 through 19 are flow diagrams illustrating the operation of an exemplary Zenu™ UDI system and method according to an embodiment of the present invention. For ease of explaining this example, the Zenu™ UDI comprises an application executing on a personal computer in a Microsoft® Windows environment. The Zenu™ UDI from a file resident in the computer system, such as in a fixed drive or other memory medium.

[0137] Turning to FIG. 12, a step 1202 represents launching of the Zenu™ UDI. Thus, once launched in this manner, the Zenu™ UDI is loaded in the computer’s random access memory (RAM) and either appears as an icon in the Window’s system tray, or is displayed for the first time. At a step 1204 available themes and templates are enumerated according to their associated file names so as to create a main UDI window, as shown at a step 1206. As part of the launching process, a decision is made at a step 1208 as to whether a default Hot Key has been overridden. If not, a default Hot Key is assumed as shown at a Step 1210. If the default Hot Key was overridden, the system will use the Hot Key override as shown at a step 1212. Next, at a decision step 1214, it is determined whether the UDI is to be shown at startup. If YES, the UDI is displayed, as shown generally at a step 1216. Otherwise, the UDI is started and placed in the Window’s system tray as an icon, as shown at a step 1218. The system then waits for an event, shown generally at a step 1220.

[0138] FIG. 13 is a flow diagram representing further details of step 1206 (create main UDI window) of FIG. 12. FIG. 13 illustrates how the Zenu™ UDI obtains its appearance and what it does in order to render such appearance using system windows from Microsoft® Windows. Creating basic windows for the UDI begins at a start step 1302. Then a set of application resources are passed to a Template .dll, as a step 1304. The Template .dll comprises the resources within the UDI that gives the Zenu™ UDI its shape and controls position of the buttons. At a next step 1306, a Theme .dll is loaded, as specified in an .ini file, or the like. The load Theme .dll stores all of the individual overrides of the Template in terms of its default appearance. At a next step, 1308, the relevant “UDI window look” is applied, together with other window settings. The details of step 1308 are described below in connection with FIG. 14. Next, at a step 1310, the buttons of the Zenu™ UDI are created as basic window definitions. The details of button creation are described below in connection with FIG. 15. Various settings of the buttons are then applied at a step 1312. The details of the button settings are described below in con-