

DeskTool

For Windows 95 and Windows NT 4.0

Version 1.7

DeskTool is a desktop toolbar designed to give you easy access to frequently used applications by displaying an organized row of buttons from which to choose. DeskTool is tightly integrated into the new shell of Windows 95 and Windows NT 4.0. Versions of Windows NT prior to version 4.0 and versions of Windows prior to Windows 95 are not supported.

DeskTool has some nifty features that you might have come to expect from a modern Windows application. Drag and Drop is supported for adding and removing buttons as well as moving buttons around on the toolbar. You can “dock” DeskTool to any one of the 4 edges of your screen, much like the Windows 95 taskbar, or leave it free floating on the desktop. You can make the toolbar “auto-hide” when docked to an edge, just as you can with the taskbar. You can also make DeskTool stay on top of other windows, making it readily accessible.

Aside from the buttons for starting applications there is a command box available that can be used to quickly issue commands to the operating system. The command box is synonymous to the “Run” dialog in the Start menu and you can issue the same commands as you can from the “Run” dialog. For example, you can start programs by simply typing their names, assuming you specify the full path to them or they can be located in your search path. You can also issue special shell commands like a drive letter and directory to get a graphical view of that directory, such as: C:\WIN95.

When you first start DeskTool a small window containing a single edit box appears free floating on your screen. This is DeskTool in its standard non-docked form. You’ll notice that you can drag the toolbar around by clicking the titlebar and moving the mouse. To dock the toolbar just drag it to the edge you want it docked to. When DeskTool is docked you’ll notice that the title bar disappears and the toolbar will appear to stick to the side of the desktop. To undock it just click somewhere outside of any buttons and the edit box and drag it away from the edge.

Adding and Moving Buttons

At this point you need to start adding the buttons to the toolbar. The simplest way to do this is to open the Windows Explorer and find an application (or a shortcut to it) that you want to add. Just drag that file onto DeskTool and a new button will be created for you. Now you can click the newly created button to start that application. New buttons always show up at the end of the toolbar, just before the command box. Once you have more than one button you can move them around by holding the Alt key down and dragging the button to where you want it. Note that if you are not holding down the Alt button when you click it will start the application! To delete a button hold down the Alt key and drag the button off the toolbar, it will then disappear.

To add a spacer between buttons Shift double click on a button. The spacer will be inserted before the clicked button. Spacers are useful for logically grouping a set of buttons. Currently spacers can only be removed using the Properties dialog.

Another way of adding a new button is to Ctrl double click in some blank part of the toolbar. You will then see the Insert Toolbar Button dialog appear. Fill in all the fields, using the “Target” and “Start In” buttons to browse for the application and start directory where appropriate, and press OK. A new button will appear at the end of the toolbar. To edit the properties of an existing button you can Ctrl double click on the button. The Edit Toolbar Button dialog will appear with the fields filled in with the current properties.

To pass parameters to an application you can specify them on the “Target” line but you’ll need to enclose the parameters in double quotes. Also make sure any parameters are specified after the target itself. As an example:

C:\Program Files\Accessories\WordPad.exe "C:\Win95\Desktop\TextFile.txt"

If you are passing parameters that contain spaces and you normally need to enclose the parameters in quotes you may continue to do so. In this case you might have 2 sets of double quotes in your parameter specification. As an example:

S:\CCMAIL\CCWIN2\WMAIL.EXE "S:\CCMAIL "Brett Robichaud" PASSWORD"

In the above example you can see that one of the parameters includes a space (the name) so it is itself enclosed in double quotes as well as the whole parameter specification.

Another method of specifying parameters is to create a Windows "Shortcut" to the application. Using shortcuts you can easily specify parameters. You can then point one of your buttons at this shortcut and you're all set! See your Windows documentation for more information on shortcuts.

Context Menu Options

To get a context menu of possible DeskTool actions you can right-click on the toolbar (anywhere other than the command box). This menu allows you to Exit the toolbar, see the About box, edit the Properties of the toolbar and any buttons, Refresh the button icons and toggle the Auto Hide status. The following is a description of each item:

The **Refresh Icons** menu item is used to retrieve the current icon from the applications represented by the toolbar buttons. For performance reasons DeskTool is designed to cache the application icons by default (this can be overridden with the Verify Targets option). There are rare times when the icon being displayed can become out of date, such as when you update an application with a new version. To make sure that the icons being displayed by DeskTool are current, select this option. It may take a few seconds as it runs around your disk getting the latest icons from your applications.

The **Auto Hide** menu item allows you to toggle the Auto Hide option on and off. You can also toggle it using the Properties dialog (described below), this is simply a quick way to toggle it. A check mark will appear next to it when the option is enabled.

The **About** menu item displays a simple dialog that documents the version number and copyright information.

The **Exit** menu option shuts down DeskTool, saving all button and state information. Keep in mind that when you properly shut down Windows DeskTool will shut itself down gracefully and save all pertinent information. Should you simply shut your machine off without shutting down Windows, or should Windows crash, DeskTool will not be able to gracefully shut down. In this situation you may lose the modifications you have made since you last started DeskTool.

Properties Dialog

The **Properties** menu option brings up a dialog that has multiple "tabbed" pages. This dialog is useful for changing the way DeskTool operates, and for modifying the buttons that exist on the toolbar.

Settings Tab

The Settings tab has a number of options that alter the way DeskTool operates. The first groupbox, labeled "**Display Settings**", contains some overall options.

The **Enable Command Box** option should be enabled if you would like the command box to appear in the toolbar. This is enabled by default.

The **Enable Splash Screen** option can be disabled if you do not want to see the humble copyright information that appears each time you start DeskTool. This is enabled by default.

The **Always on Top** option forces the toolbar to stay on top of all other windows. Any window that might normally overlap it will end up underneath the toolbar. This makes DeskTool readily available whenever you need it. This is enabled by default.

The **Verify Targets** option forces DeskTool to go out to each application and get its icon each time you start DeskTool. This slows down the startup of DeskTool in direct relation to the number of buttons you have added, by disabling the caching of application icons. Typically this option should be disabled to get the best startup performance from DeskTool. This is disabled by default.

The groupbox labeled “**Toolbar Button Size**” has two options: Small buttons and Large buttons. Choose one of these to determine how big the toolbar buttons will appear. Try them both to get a feel for which is easier for you so use. Small buttons are enabled by default.

The “**Auto Hide Settings**” groupbox allows you to enable the Auto Hide feature. Auto Hide is only relevant to docked toolbars and allows the toolbar to slide off screen when not being used. To gain access to the toolbar after it disappears just move the mouse cursor all the way to the edge that the toolbar is docked to and it will reappear. Once you’ve shifted focus back to another application the toolbar will magically disappear again. One limitation of the Auto Hide feature is that only one toolbar with this setting can exist on any edge. You can have multiple toolbars on an edge, but only one can be Auto Hidden, including the taskbar. You can also set the speed at which DeskTool will slide off the screen when Auto Hide is enabled. Simply adjust the slider control to make it disappear and reappear faster or slower. Experiment with this control to see what works best for you.

The “**Tooltip Settings**” groupbox allows you to enable the Tooltip feature. ToolTips are small popup text boxes that show up when you hold the mouse cursor over a button (without clicking). The text that appears in the popup is the title text you set for each button. ToolTips are a great feature that makes it much easier to determine what a button does, without it you have to recognize each icon by picture only. You may also set the speed at which the tooltip appears by adjusting the slider control. ToolTips are enabled by default.

Finally the “**To Start Application**” groupbox allows you to choose how DeskTool acts when you click on a button. When set to the Single click option DeskTool will start the application each time you click on a button. When set to the Double click option nothing will happen until you double click on a button.

Buttons Tab

The other tab in the Properties dialog is labeled “Buttons”. From here you can insert/edit/remove and move any of the buttons you have defined.

The main control in this tab is a large list that shows you each of the buttons you currently have defined along with its icon. The list is displayed in the order the buttons are currently shown on the toolbar. To the immediate right of this listbox are two arrow buttons. These are used to move the currently selected button around on the toolbar. This is synonymous with Alt dragging the buttons around the toolbar. Go ahead and play with these buttons, you’ll see the button on the toolbar itself move as you use them.

Further to the right are buttons for inserting, editing and removing toolbar buttons. The **Insert** and **Edit** buttons will bring up the familiar Insert and Edit dialogs mentioned previously. Using these are synonymous with Ctrl Double clicking in the toolbar itself. The **Remove** button simply removes the currently selected button in the list and is synonymous with Ctrl dragging a button off the toolbar.

The final button labeled **Insert Space** is used to add a space between buttons. This is useful for grouping related buttons on the toolbar. You may use multiple spacers to widen the gap between groups. You can also add spacers by Shift double clicking a button on the toolbar, a spacer will be inserted before the clicked button. Note that there currently is no way to remove spacers directly from the toolbar, you must use this properties dialog to do this.

Using Multiple Toolbars

Under normal operation DeskTool writes its data to the file DeskTool.CFG within the same directory as DeskTool.EXE. Because of this running multiple copies of DeskTool requires a special command line option. Using `-d{name}` on the command line when starting DeskTool (where {name} is some 8 character or less string) allows you to name each instance of DeskTool.

For instance, say you wanted to have 2 copies of DeskTool running at one time. On one toolbar you have your development tools while the other has applications. What you would want to do is setup 2 shortcuts, one to run DeskTool with the option `-dDevelop` and the other with `-dApps`. Do this by specifying the following text within the quotes in the Target editbox of the Shortcut:

“desktool -dDevelop”

Specify a directory where necessary. Then start DeskTool using these new shortcuts instead of running DeskTool directly.

You can create as many DeskTool setups as you like as long as they have unique names.

Assigning a Custom Icon to a Button

DeskTool itself does not yet support custom icons for its buttons (this is a highly requested feature that will be implemented someday). However, there is a simple workaround to this missing feature. Since shortcuts (officially known as Shell Links or .lnk files) allow you to assign an icon from any file, you can use a shortcut to do the dirty work for DeskTool. Start by creating a shortcut for the app in question. Select the desired icon in the Shortcut page of its properties and drag that shortcut onto DeskTool. You should now magically see the desired icon!

Some system created shortcuts, such as for Internet Explorer or MS Mail & News readers do not by default have the icon assigned to them. Sometimes when you drag these shortcuts onto DeskTool you will not see the desired icon. To fix this, see the section on Known Problems below.

Known Problems

1. When using the Auto Hide feature ToolTips may not work correctly in other applications. If this is a problem for you simply disable the Auto Hide feature of DeskTool.
2. DeskTool does not detect screen resolution changes on the fly. You have to exit then restart DeskTool before it will recognize the change.
3. When pointing to shortcuts (also known as Shell Links or .lnk files) sometimes you will see the wrong icon appear. If this happens open the properties for that link file (right click on it and choose properties) and choose the Shortcut tab. Now click the Change Icon button. When the dialog appears choose the icon you want to use and click OK. Even if it appears that the icon you want is selected make sure you click OK (the icon is not really fully selected so you need to click OK). Now refresh icons in DeskTool and you should see the correct icon.

Release History

Version 1.7 (4/20/97) – A number of annoying problems have been addressed with this release, including:

- Icon handling has been dramatically improved, particularly small icons. Choose Refresh Icons from the context menu and your small icons should improve their appearance.
- Also fixed a long-standing bug where the links in shortcut files were not being properly resolved. This resulted in the wrong icons often appearing for .lnk files.
- There is now a documented workaround for assigning a custom icon for a button. See the new section titled “Assigning a Custom Icon to a Button” for details.
- A rather nasty little resource leak was plugged with this release.
- Ctrl double clicking a button that resides to the left of a spacer no longer pushes that button to the right side of the spacer.
- Ctrl dragging buttons has been replaced with the more standard Alt dragging.
- The drag positioning logic has been greatly enhanced. It is now possible to place a button on either side of an existing group of spacers. Previously the button would only position to the right of the spacer group.
- You can now insert a spacer by Shift double clicking a button. The spacer will be inserted before the clicked button. Currently the only way to delete spacers is in the Properties dialog.

Version 1.6 (4/6/97) – Now includes full install/uninstall capabilities! Switched back to using statically linked MFC libraries to minimize download size. Adding the DLLs in version 1.51 increased the download size by 250%. Version 1.6 is available in full install/uninstall version or a much smaller EXE only zipfile.

Version 1.51 (3/9/97) – Recompiled with Visual C++ 5.0 (reduced binary size by 12%). Changed to using shared MFC DLLs to further reduce binary size and promote code sharing. Limited release.

Version 1.5 (5/30/96) – Feature complete! Everything is implemented and working, with a few known limitations. Wide availability (all of Tally).

Version 1.0 (2/15/96) – The first release (obviously!). Mostly complete but some functionality not yet implemented. Limited release (Gary?).

Contact Information

DeskTool is a freeware application authored and copyrighted by:

RobiSoft
Brett Robichaud
51 Village Green
West Lebanon, NH 03784
Brett@ma.ultranet.com

Please send any comments or suggestions to the above email address.

While this is freeware I certainly do accept donations. Any amount that you feel is deserving would be much appreciated to help fund future utilities. Please send any donations to the above address.

Distribution

You are hereby licensed to make as many copies of this software and documentation as you wish; give exact copies of the original version to anyone; and distribute the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SUPPLIED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED.

NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedures dictate that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.