# VirtuaWin v4.4 Help

## **Overview**

VirtuaWin is a virtual window manager for Windows Win9x/NT/2000/XP/Vista/7 with up to twenty desktops. It's quick and easy to use with no fancy graphics, just an icon in the system tray indicating current desktop.

VirtuaWin is Copyright (c) 1999-2005 Johan Piculell, Copyright (c) 2006-2012 VirtuaWin (VirtuaWin@home.se) and is licensed through the GNU General Public License. Read the COPYING.TXT file for the complete license.

We take no responsibility for data losses or damage caused by VirtuaWin.

For latest version, question forum and bug reporting please visit <a href="http://virtuawin.sourceforge.net/">http://virtuawin.sourceforge.net/</a>.

## **Functions**

These are the basic functions of VirtuaWin:

- Use the configured keys and or mouse to switch to a new desktop.
- Left double click on the system tray icon to bring up the setup dialog or right click and choose setup to show the <a href="Setup Dialog">Setup Dialog</a>.
- Left click on the icon to show the window list, click on a window title that you want to keep on all desktops (sticky). There is optional usage of this list, see <a href="Setup Dialog">Setup Dialog</a>.
- Middle click or Shift middle click on the icon to move forward or backward a desktop.
- Right click and choose gather to make all windows visible on the current desktop.
- If the system tray icon flashes after a desktop switch, this means that an application did not respond and VirtuaWin was not able to hide/show it, the window is not lost, it will be recovered once the window starts responding.

## **Install/Uninstall**

The installation and un-installation of VirtuaWin is fully automated.

To install run VirtuWin\_setup\_<version>.exe or, if wide character support is required, run VirtuWin\_Unicode\_setup\_<version>.exe. For uninstalling, quit VirtuaWin and choose the uninstall icon, however, if you add custom icons or modules, the un-install process will not be able to remove the "icons" or "modules" folder – this must be done manually.

VirtuaWin does not add or require any registry keys to run, only the required uninstall program registry keys are added by the installer. Nothing else is added to your system, all VirtuaWin files are placed in the installation directory of VirtuaWin.

#### **Control Menu**

The Control Menu can be opened by:

- Right clicking on the system tray icon.
- Binding the *UI: Open standard control menu* command to a hotkey and pressing this hotkey.
- Binding the *UI: Open compact control menu* command to a hotkey and pressing this hotkey.

The control menu is used to access main VirtuaWin functionality, the menu contains the following items:

- *Setup*: Opens the main VirtuaWin <u>Setup Dialog</u>.
- Window Rules: Opens the Window Rules Dialog.
- *Reapply Rules*: Re-initializes all windows visible on the current desktop, applying all window rules appropriately. Window settings like 'Always show' will be reset.
- Gather All: Moves all windows to the current desktop.
- *Help*: Opens this help documentation.
- *Enable* or *Disable*: Toggles the current active state of VirtuaWin, if Disabled is selected all VirtuaWin functionality is disabled except for access to the Control Menu so VirtuaWin can be re-enabled.
- *Exit VirtuaWin*: Exits VirtuaWin; VirtuaWin will gather all windows before exiting so that no application is lost.
- *Move to Desk* #: Moves to the selected desktop; the compact form of the control menu puts these items into a sub menu.
- *Move to Next*: Moves to the next desktop.

• *Move to Previous*: Moves to the previous desktop.

## **Window Menu**

The Window Menu can be opened by:

- Clicking the middle mouse button on the window title bar, see <u>Setup Dialog</u>'s Enable middle button window menu activation setting.
- Binding the *UI: Open standard window menu* command to a hotkey and pressing this hotkey while a window has input focus.
- Binding the *UI: Open compact window menu* command to a hotkey and pressing this hotkey while a window has input focus.

The window menu is used to control and configure the current window, i.e the window currently with the input focus. The menu contains the following items:

- **Dismiss Window**: If the current window is only being temporarily show (for example the window list 'Show' feature was used to display it, also see option On hidden window activation) it will be moved back to its original desktop, otherwise the window will be minimized.
- *Push to Bottom*: Pushes the window to the back of the desktop, i.e. under every other window.
- *Always on Top*: Toggles the Window 'Always on top' property which keeps the window above other windows.
- *Manage Window*: Forces VirtuaWin to manage the current window, this item is only available if the window is not currently being managed.
- *Stop Managing*: Forces VirtuaWin to not manage the current window, this item is only available if the window is currently being managed.
- *Always Show*: Toggles whether the current window is visible on all desktops or just the current one.
- *Gather*: moves all the windows created by the current window's process onto the current desktop.

- *Move to Next*: Moves the current window to next desktop; this command will fail if the current desktop is the last desktop and Enable desktop wrapping is disabled.
- *Move to Previous*: Moves the current window to next desktop; this command will fail if the current desktop is the first desktop and Enable desktop wrapping is disabled.
- *Move to Desk* #: Moves the current window to the selected desktop; the compact form of the window menu puts these items into a sub menu.
- *Edit Window Rule*: Opens the Window Rule Dialog with the rule being applied to the current window selected, this item is only available if the current window had a rule applied to it.
- *Add Window Rule*: Opens the Window Rule Dialog with the window identification section filled with the current window's details (i.e. class name etc.), normal only one name is required for identifying the window type so the others should be removed.

**Note:** The new rule will not be created until you Add the rule.

• *Info*: Opens a dialog giving detailed information about the current window.

**Note:** The text in this dialog can be copied using standard Windows Control+C key.

#### **Window Lists**

#### Standard Window List

The Standard Window List can be opened by:

- Clicking the left mouse button on the system tray icon if Setup option
   <u>Mouse opens compact window list</u> is disabled, or by clicking the left mouse
   button on the system tray icon with the Shift key pressed if this option is
   enabled.
- Clicking the middle mouse button on the desktop if Setup option <u>Enable</u> <u>middle button window list activation</u> is enabled, the Shift key must be pressed if <u>Mouse opens compact window list</u> is enabled.
- Binding the *UI: Open standard window list* command to a Hotkey and pressing this Hotkey.

The standard window list can be configured using Setup's <u>Available window list</u> actions to display the following columns:

- **Switch To**: VirtuaWin will change the current focus to the window selected form this column, if the window belongs to a different desktop to the current VirtuaWin will also change desktops.
- *Move Here*: A window selected from this column will be moved to the current desktop and given the focus; only windows not on the current desktop will be listed in this column.
- *Show*: A window selected from this column will be shown on the current desktop, but it is not moved; this means the window can be <u>dismissed</u> back to its original desktop.
- *Always Show*: Toggles whether a window is shown on all desktops, the windows icon is replaced with a tick icon if the window is currently shown on all desktops.

Only the columns enabled in the *Available window list actions* are initially

displayed, however all columns will be displayed if a column title is clicked on.

**Note:** This window list form is restricted to a maximum of one screen depth of windows, if more windows exist than can be displayed VirtuaWin will automatically use a compact window list.

## **Compact Window List**

The Compact Window List can be opened by:

- Clicking the left mouse button on the system tray icon if Setup option
   Mouse opens compact window list is enabled, or by clicking the left mouse
   button on the system tray icon with the Shift key pressed if this option is
   disabled.
- Clicking the middle mouse button on the desktop if Setup option <u>Enable</u> middle button window list activation is enabled, the Shift key must be pressed if <u>Mouse opens compact window list</u> is disabled.
- Binding the *UI: Open compact window list* command to a Hotkey and pressing this Hotkey.

As with the *Standard Window List* form the compact window list can be used to display 4 different columns. The compact form only displays one column at a time, click on the title to change to the next column (or the hotkey 'n'). The compact window list will first cycle through the columns enabled in Setup's <a href="Available window list actions">Available window list actions</a> and then cycle through all columns.

Unlike the standard window list, the compact window list can display more than one screen depth of windows and displays more of each window's title bar text.

#### **MRU Window List**

The Most Recently Used (MRU) Window List can be opened by:

- Clicking the left mouse button on the system tray icon with the Control key pressed.
- Clicking the middle mouse button on the desktop with the Control key pressed

if Setup option **Enable middle button window list activation** is enabled.

• Binding the *UI: Open most recently used window list* command to a Hotkey and pressing this Hotkey.

The MRU window list displays a single list of windows, order by how recently they they last had the input focused (i.e. similar to the main Windows Alt+tab window list). VirtuaWin will change the current focus to the window selected from this list, VirtuaWin will also change desktops if the window belongs to a different desktop to the current.

# **Setup Dialog**

The Setup dialog can be opened by:

- Double clicking the left mouse button on the system tray icon.
- Selecting *Setup* in the <u>Control Menu</u>.
- Binding the *UI: Open VirtuaWin setup* command to a Hotkey and pressing this Hotkey.
- Running the VirtuaWin program while VirtuaWin is already running.

The setup dialog consists of 5 different setup tabs with keys, mouse, modules and other various configuration options under them. It is possible to use the *Apply* button to try out different settings. If *Cancel* is pressed, all changed from the last apply will be undone.

#### **General Tab**

#### **Desktop Configuration**

- With *Desktop Layout* you can configure how many desktops you want and how they shall be positioned. You can have up to 20 desktops. If you for example set this to 8 wide by 1 deep, you will get a horizontal row of eight desktops. The "original" icon will be used if you set 2 by 2, otherwise numbers will be used. See also <u>Custom Icons</u>. Please note that if you set the size to fewer desktops any windows on the removed desktops will be automatically moved to desktop 1.
- *Enable desktop wrapping*: If checked, it will be possible to cycle through all desktops in both directions, meaning that from the last desktop in any direction, it's possible to go to the first again by continuing in the same direction.
- *Name of desktop* # defines the name of the current desktop, this is used in the tooltip given when the mouse is moved over the system tray icon. If no name is give the name used will be 'Desktop #' where # is the current desktop number.

• The *Next* button can be used to move to the next desktop.

#### **User Interface**

- **Available window list actions** can be used to configure the contents of the Window List menu, available options are:
- *Switch To* can be used to access any window directly, meaning that VirtuaWin will change to the window's desktop and give the focus to the window.
- *Move Here* is used for moving the selected window to the current desktop, useful for moving windows to different desktops.
- *Show* is used to temporarily display the selected window on the current desktop; the window will be returned to its original desktop when it is either dismissed (see the <u>dismiss window</u> hotkey) or on the next desktop change.
- *Always Show* is used to toggle whether a window is visible on all desktops (sometimes termed 'sticky').

It is possible to choose any combination and if more than one function is selected, the menus will be displayed side-by-side. Selecting a column title, e.g. click on '**Switch To**', will change the menu to display all columns.

**Note:** The menus will only contain the windows that are possible to affect during the time, for example, moving a window to the same desktop that it already belongs to is pointless and therefore windows on the current desktop will not appear in the **Move Here** column.

- *Use desktop title lines in window list*: If checked each group of windows on a desktop within the Window List menu will be headed by a desktop title line, making the menu easier to read. The title line also displays the desktop name if one has been set.
- *Mouse opens compact window list*: If checked the compact form of the window list is opened by the mouse rather than the standard form. The window list can be opened using the mouse by left clicking on the system tray icon or by middle clicking on a the desktop (if enable, see <a href="Enable middle button window">Enable middle button window</a>

#### <u>list activation</u>).

The compact form of the Window List displays only one column, clicking on the title of the column will change the menu to the next column.

**Note:** The compact list has two further advantages, firstly it is able to support many more windows (multiple menu columns will be automatically used if there are too many windows to display in a single column) and the length of text for each window is truncated to 128 characters rather than only 30 for the noncompact form.

• *Mouse opens compact window menu*: If checked the compact form of the window menu is opened by the mouse rather than the standard form. The window menu can be opened using the mouse by middle clicking on a the desktop (if enable, see <u>Enable middle button window menu activation</u>).

**Note:** The compact form of menus use sub-menus to keep the size of the main menu small.

- *Mouse opens compact control menu*: If checked the compact form of the control menu is opened by the mouse rather than the standard form. The control menu can be opened using the mouse by right clicking on the system tray icon.
- *Hotkey opens menu or list at fixed location*: If enabled VirtuaWin will open all menus and dialogs at a fixed location on the screen, if disabled most menus and dialogs are opened next to the mouse.

#### **Hotkeys Tab**

The Hotkeys tab is used to bind system wide 'hotkey' combinations to VirtuaWin functionality, as the hotkey is system wide a key combination must not be used by any other application including Windows itself. The following commands can be bound to a hotkey:

- *NAV: Move left*: Moves to the next desktop to the left, this command fails if the current desktop is in the left most column and <u>Enable desktop wrapping</u> is disabled.
- *NAV: Move right*: Moves to the next desktop to the right, this command fails if the current desktop is in the right most column and <u>Enable desktop wrapping</u>

is disabled.

- *NAV: Move up*: Moves to the next desktop above (which desktop this is depends on your Expert <u>Invert Up/Down</u> setting). This command fails if the current desktop is on the top row and <u>Enable desktop wrapping</u> is disabled.
- *NAV: Move down*: Moves to the next desktop below (which desktop this is depends on your Expert <u>Invert Up/Down</u> setting). This command fails if the current desktop is on the bottom row and <u>Enable desktop wrapping</u> is disabled.
- *NAV: Move to next*: Moves to the next desktop; this command will fail if the current desktop is the last desktop and <u>Enable desktop wrapping</u> is disabled.
- *NAV: Move to previous*: Moves to the previous desktop; this command will fail if the current desktop is the first desktop and <u>Enable desktop wrapping</u> is disabled.
- *NAV: Move to desktop* #: Moves directly to the desktop specified.
- *NAV: Return to last desktop*: Moves to the last desktop. Note the last desktop is defined to be the last one you stayed on for more than a second and isn't the current one.
- *WIN: Toggle always show*: Toggles whether the current window is visible on all desktops or just the current one.
- *WIN: Always on top*: Toggles the Window's 'Always on top' property which keeps the window above other windows.
- *WIN: Bring to top*: Brings the window under the mouse to the top of the desktop giving it the current focus. Note that this does not set the Window's 'Always on top'.
- *WIN: Push to bottom*: Pushes the window to the back of the desktop, i.e. under every other window.
- *WIN: Dismiss window*: If the current window is only being temporarily show (for example the window list '**Show**' feature was used to display it, also see option On hidden window activation) it will be moved back to its original desktop, otherwise the window will be minimized.

*WIN: Gather process windows*: Moves all the windows created by the current window's process onto the current desktop.

- *WIN: Move left*: Moves the current window to the next desktop to the left; this command will fail if no window currently has focus or if the current desktop is the left most desktop and <u>Enable desktop wrapping</u> is disabled.
- *WIN: Move right*: Moves the current window to the next desktop to the right; this command will fail if no window currently has focus or if the current desktop is the right most desktop and <u>Enable desktop wrapping</u> is disabled.
- *WIN: Move up*: Moves the current window to the next desktop above (which desktop this is depends on your Expert <u>Invert Up/Down</u> setting); this command will fail if no window currently has focus or if the current desktop is the top most desktop and <u>Enable desktop wrapping</u> is disabled.
- *WIN: Move down*: Moves the current window to the next desktop below (which desktop this is depends on your Expert <u>Invert Up/Down</u> setting); this command will fail if no window currently has focus or if the current desktop is the bottom most desktop and <u>Enable desktop wrapping</u> is disabled.
- *WIN: Move to next desktop*: Moves the current window to next desktop; this command will fail if no window currently has focus or if the current desktop is the last desktop and Enable desktop wrapping is disabled.
- *WIN: Move to previous desktop*: Moves the current window to next desktop; this command will fail if no window currently has focus or if the current desktop is the first desktop and <u>Enable desktop wrapping</u> is disabled.
- *WIN: Move to desktop* #: Moves the current window to the specified desktop; this command will fail if no window currently has focus.
- *WIN: Move left and follow*: Same as *WIN: Move left* except VirtuaWin changes to desktop as well.
- *WIN: Move right and follow*: Same as *WIN: Move right* except VirtuaWin changes to desktop as well.
- *WIN: Move up and follow*: Same as *WIN: Move up* except VirtuaWin changes to desktop as well.

- *WIN: Move down and follow*: Same as *WIN: Move down* except VirtuaWin changes to desktop as well.
- *WIN: Move to next desktop and follow*: Same as *WIN: Move to next desktop* except VirtuaWin changes to desktop as well.
- *WIN: Move to prev desktop and follow*: Same as *WIN: Move to previous desktop* except VirtuaWin changes to desktop as well.
- *WIN: Move to desktop # and follow*: Same as *WIN: Move to desktop #* except VirtuaWin changes to desktop as well.
- *WIN: Gather all windows*: Moves all windows being managed by VirtuaWin onto the current desktop.
- *WIN: Reapply window rules*: Re-initializes all windows visible on the current desktop, applying all window rules appropriately. Window settings like 'Always show' will be reset.
- *UI: Open standard control menu*: Opens VirtuaWin's standard <u>Control Menu</u>, which can also be opened by right clicking on the system tray icon.
- *UI: Open compact control menu*: Opens VirtuaWin's compact <u>Control Menu</u>, which can also be opened by right clicking on the system tray icon. The compact menu moves some menu items into sub-menus ensuring the size of the menu does not get unmanageable.
- *UI: Open standard window menu*: Opens the standard <u>window menu</u> to control the current window; this command will fail if no window currently has focus.
- *UI: Open compact window menu*: Opens the compact <u>window menu</u> to control the current window; this command will fail if no window currently has focus. The compact menu moves some menu items into sub-menus ensuring the size of the menu does not get unmanageable.
- *UI: Open standard window list*: Opens the standard window list menu, which is used to access any window.
- *UI: Open compact window list*: Opens the compact window list menu which

is used to access any window. See option <u>Mouse opens compact window list</u> for more information on the differences between the two window list forms.

- *UI: Open most recently used window list*: Opens a menu listing all windows in order of last use, VirtuaWin will switch to the selected window.
- *UI: Open window rules setup*: Opens the Window Rules Dialog.
- *UI: Open VirtuaWin setup*: Opens VirtuaWin's main <u>Setup Dialog</u>.
- *UI: Toggle enable state*: Toggles whether VirtuaWin is enabled or not; when VirtuaWin is disabled this hotkey command is the only one which remains bound.
- *UI: Toggle display of system tray icon*: Toggles the display of the VirtuaWin icon in the taskbar system tray.
- *UI: Boss key*: Hides sensitive application windows by moving immediately to desktop 1 and then disables & hides VirtuaWin itself. If an <u>Unboss key</u> is defined then only the *unboss key* can be used to restore access to VirtuaWin and the other desktops, if no *unboss key* is defined then the *boss key* acts as a toggle, i.e. press once to active boss key mode and press again to restore VirtuaWin.

**Note:** Boss key takes the highest priority, so even if VirtuaWin is currently disabled (see <u>Toggle enable state</u>) pressing this key will still make VirtuaWin change to desktop 1 and hide. It cannot however disable modules such as the numerous pagers which run independently.

- *UI: Boss key*: Deactivates Boss key mode see <u>Boss key</u> above.
- *UI: Exit VirtuaWin*: Exits VirtuaWin.

#### **Hotkeys**

- The first sub-window on the Hotkeys tab contains a list of current hotkeys, select a line to *Modify* or *Delete* the selected hotkey. When a line is selected, the remaining fields in the tab are updated to reflect the hotkey selected.
- Command is used to configure what command is to be executed by the hotkey,

see above for a complete list of available commands and what they do.

• **Desktop**: Some commands require a desktop to be specified, e.g. **NAV: Move to desktop** #, the Desktop contains a drop down list of all available desktops, use this to specify the required desktop.

**Note:** Usually this list only contains normal visible desktops, if using a hidden desktop you must first go to the hidden desktop for it to appear in this list.

- *Operate on window under mouse*: This option is available for commands which operate on a 'current' window. If enabled the current window is defined to be the one which is currently under the mouse cursor rather than the one which has the current input focus.
- Use the *Hotkey* entry to specify the required hotkey by simply clicking into the entry and pressing the required Hotkey combination, the + *WIN key* checkbox will also need to be set depending on whether the Windows key modifier is desired.

**Note:** If the Hotkey is already bound by something else (including VirtuaWin and Windows) the entry will not be correctly changed, it may be set to **None** and the action bound to the hotkey will be executed. This is a limitation of the Windows Hotkey interface, see the main VirtuaWin FAQ for instructions on how to manually configure some hotkeys.

- Pressing the *Add* button add the current Command/Hotkey configuration as a new hotkey, it will be added to the bottom of the list. This button is not enabled if the Hotkey is not set to a valid hotkey combination.
- Pressing the *Modify* button up-date the currently selected hotkey with the current Command/Hotkey configuration. This button is not enabled if a hotkey is not selected in the list or the Hotkey is not set to a valid hotkey combination.
- Pressing the *Delete* button removes the currently selected Hotkey. This button is not enabled if a hotkey is not selected in the list.

#### **Mouse Tab**

## **Mouse Support**

- *Enable mouse desktop changing*: If checked, it will be possible to change desktop and drag windows to other desktops with the mouse.
- *Only change when dragging a window*: If checked, desktop changing with the mouse is only possible when dragging a window.
- The *Delay before changing desktop* slider configures how long it will take to switch desktop when the mouse goes to the ends of the screen. You can set a value between 50 4000 ms.
- *Enable modifier key control*: If checked, you can choose a key that must be held down if the mouse should move to a new desktop. This is useful for avoiding accidental desktop switch when using Mouse support.
- *Enable desktop edge knocking*: If checked the mouse must 'knock' the desktop edge before VirtuaWin will change desktops. To knock the edge the mouse must move to the desktop edge, then move away from the edge (by a quarter of the Mouse jump length), before going back to the edge this action must be performed within a second. So if the Mouse jump length is set to 60 (recommended) the mouse must move 15 pixels away from the edge, also the mouse must not wander in the direction perpendicular to the edge by more than the Mouse jump length.
  - When *Always required* is also checked, a knock must always be performed to change desktops with the mouse. If disabled a knock is only required when the mouse operation starts near the edge (within 'Mouse jump length', e.g. 60 pixels). For example, if the <u>modifier key control</u> is pressed within 60 pixels of the edge a knock will be required, if its pressed outside this boarder no knock will be required.
- The *Mouse jump length* field tells how many pixels the mouse pointer will jump away from the desktop edge on a mouse desktop change. Values between 0-999 are possible, but don't set this value any higher than the resolution of your screen. A value of at least 10 is recommended to avoid unwanted double desktop changing and a value of around 60 or more if edge knocking is enable.
- *Enable mouse warping*: If checked, the mouse will be moved to the opposite side of the screen when going over the edge. The effect of the Mouse jump field is the same.

- *Enable middle button desktop changing*: If checked, clicking the middle mouse button on the desktop and moving the mouse will change desktops in the direction of the mouse movement. The size of the mouse movement required to change desktops is a quarter of the <u>Mouse jump length</u>.
- *Enable middle button window list activation*: If checked, clicking the middle mouse button on the desktop will open a VirtuaWin window list (see *Window List Contents* section below). This menu can also be opened using a hotkey, see *Enable window list hotkey*.
- *Enable middle button window menu activation*: If checked, clicking the middle mouse button on a window's title bar will open a VirtuaWin control menu for the window. This menu can be used to dismiss the window and control which desktop it is placed on. This menu can also be opened using a hotkey, see <u>UI: Open standard window menu</u>.

#### **Modules Tab**

The original installation of VirtuaWin includes only one module, WinList module. More info about modules is available on <a href="http://virtuawin.sourceforge.net/">http://virtuawin.sourceforge.net/</a>. There you can also find more modules to download. There is currently a max limit of 10 modules that VirtuaWin will handle.

- The list box will display the modules that are currently loaded.
- Pressing the *Configure* button, or by double clicking, will bring up the setup dialog for the selected module, if the selected module has one.
- Press the *Reload* button for unloading all modules and re-read the module directory.
- Pressing the *Enable/Disable* button does exactly what it sounds like. Note that you need to restart VirtuaWin in order for enabled modules to become active again. Disabling modules works instantly.
- Pressing the *Get Modules* button will open VirtuaWin's main modules download page in your preferred web browser.

## **Expert Tab**

#### **Advanced Options**

• On desktop change preserve: Setting this option to Static Taskbar Order will make VirtuaWin build up and maintain its own list of windows in creation order, this order will be used to order the buttons in the taskbar when changing desktops; VirtuaWin will not attempt to restore the Z order. When the Z order is selected VirtuaWin will maintain the order that windows are on top of each other, this is the same order as used by the Windows Application Menu (Alt + Tab). When used, the order of windows in the taskbar is not preserved and there is a desktop changing performance hit of around 5% compared to Taskbar Order. When Static Taskbar & Z Order is used both orders are restored but at about a 10% performance cost. The Dynamic Taskbar order settings differ from the static in that they get the taskbar button order from Explorer just before changing desktops and corrects the button order, this allows VirtuaWin to support 3rd party tools like Taskbar Shuffle which can be used to reorder the buttons in the taskbar - these options do have an additional small performance hit.

**Note:** The dynamic taskbar feature is not currently supported in Windows 7.

- On hidden window activation: A hidden window is one that currently belongs to a different desktop so it is being hidden; activation is some event that suggests to VirtuaWin that the owning application wants the window to be displayed (e.g. opening a url link may activate a hidden internet browser). Setting this option to Ignore the event which ensure that VirtuaWin will not allow the window to display. When set to Move window to the current desktop VirtuaWin will move the window to the current desktop and display the window. If set to Show window on current desktop the window will be display but it is not moved to the current desktop, instead the window will be returned to its original desktop when it is either dismissed (see the dismiss window hotkey) or on the next desktop change. When set to Move to window's desktop VirtuaWin will do a desktop change to the window's desktop, which will also display the window. This action can also be set differently for each window type using a window rule, see the Window Rule Dialog option.
- *Restore focus on desktop change*: If checked, when switching desktops the focus will be returned to the window that last had the focus on the desktop. If

disable the focus is removed - no window will have the focus.

- *Hide minimized windows on desktop change*: If checked, minimized windows will be hidden on a desktop change. If not checked, their taskbar button will remain visible on all desktops; this can be used as an alternative method for changing a window's desktop (instead of dragging with mouse or using a hotkey).
- *Invert Up/Down*: If checked, the up/down control keys and mouse behavior will be switched. This is nice to use when you have hot keys set on the numpad. For example on a 3x3 desktop setup, use hot keys 1-9 on the numpad and invert up/down. Then every desktop will be mapped to corresponding key number and the control keys/mouse will behave as expected. Hard to explain, try it out...
- *Hide system tray icon*: If checked the system tray icon will be removed. To see the setup dialog just start VirtuaWin a second time and the setup dialog will appear.

#### **Workarounds**

Options in this group should only be changed if VirtuaWin is not functioning correctly.

- *Enable taskbar detection*: During startup, VirtuaWin will try to locate the taskbar to be able to improve mouse based navigation and alternative Move window hiding. If you use an alternative shell, e.g. LiteStep, then you wont have a taskbar and probably want to disable this. If this option is disabled <a href="Dynamic taskbar order">Dynamic taskbar order</a> will not be available.
- *Enable Windows hook*: If enabled, VirtuaWin installs a system shell hook which monitors activity events (such as flashing taskbar buttons) of all windows to ensure none are missed.

**Note:** If using this option before up-grading VirtuaWin it is recommended you exit VirtuaWin 10 minutes or more before up-grading, this will reduce the likelihood of having to reboot your system.

• *Enable window specific rules*: If enabled, window specific rules created via the <u>Window Rules Dialog</u> will be applied, see help about the <u>Window Rules</u>

<u>Dialog</u> for more information. This option should be enabled, if not some windows, most notably Explorer windows, may disappear.

• *Enable dynamic taskbar button removal*: If checked and a <u>Dynamic taskbar order</u> is used VirtuaWin will use the taskbar button order returned by Explorer to remove unwanted buttons, these buttons are sometimes created by programs which create a single main window and taskbar buttons for each viewable file (e.g. Excel & Acrobat).

**Note:** The dynamic taskbar feature is not currently supported in Windows 7.

- *Force release of modifier keys on desktop change*: If checked, when VirtuaWin changes desktop it will send messages to the new current application to force it to drop any currently pressed modifier key (Alt, Control, Shift & Windows keys). Some applications fail to 'drop' these modifier keys when a hotkey is used to change desktops.
- *Force refresh after desktop change*: If checked, VirtuaWin will do a refresh after a switch to a new desktop, similar to pressing F5 with "desktop focus". Useful if applications are leaving traces after them (I have only seen this problem with PhotoShop). Otherwise, uncheck it to get rid of the annoying flicker.
- *Only hide taskbar button of minimized windows*: If checked, VirtuaWin uses the *Toolwin flag* method of hiding just the taskbar button for all minimized windows (assuming <u>Hide minimized windows on desktop change</u> is disabled), this preserves the Window's Thumbnail on Windows 7. This option has been introduced in case this change introduces unwanted side effects if no issues are reported when this option is enabled the option will be removed in the next release.
- *Write events to the virtuawin.log*: If checked VirtuaWin creates a log file detailing all VirtuaWin related events in your user configuration directory (to locate this file see <a href="Explore Config">Explore Config</a>). This is very useful when attempting to reproduce issues and reporting problems to VirtuaWin.
- Pressing the *Log Windows* button adds information about every window to the virtuawin.log file, this helps gather information required when a problem is encountered. The *Write events to the virtuawin.log* option must be enabled for this feature to have any effect.

| • Pressing the <i>Explore Config</i> button opens an explorer window displaying the contents of your VirtuaWin configuration directory; this can be used to find your configuration files. |
|--|
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# **Window Rules Dialog**

The Window Rules dialog can be opened by:

- Selecting *Window Rules* in the <u>Control Menu</u>.
- Binding the *UI: Open window rule setup* command to a hotkey and pressing this hotkey.
- Selecting *Add Window Rule* or *Edit Window Rule* in the <u>Window Menu</u>.

The window rules dialog is used to improve the way VirtuaWin handles certain types of windows. Each window rule consists of two basic parts, the first part controls the window identification, i.e. determines what windows the configuration is to be applied to, the second part configures the settings to apply to these windows. Each window can have at most one window rule applied.

- The first sub-window on the Window Rules dialog contains a list of all currently defined window rules, the order of the list is very important, as a window can only be assigned one window rule, when a new window is opened VirtuaWin starts at the top of the list trying to find a window rule that matches, it stops at the first match.
- Press the *Up* or *Down* button to move the currently selected window rule up or down the list; these buttons simply change the order of items in the list, they do not apply any modifications made to the current window rule.
- The *Add* button creates a new window rule at the top of the list; its window identification and settings will be as per the current dialog state.
- The *Modify* button applies any changes made to the current window rule's identification and settings.

**Note:** Until this button is pressed all changes to the current window rule are ignored; a common mistake is to select a window rule, make some changes and then press Okay or Apply - this has no effect as the selected window rule was not modified!

- The *Delete* button removes the currently selected window rule.
- The *OK* and *Apply* buttons will save any changes made to the Window Rules configuration (see note for Modify) and reapplies the window rules to all windows on the current desktop.

#### **Identification**

A combination of three fields is used to identify what windows the window rule is to be applied to:

- *Class Name*: Defines the name of the window's class, this can be found in the <a href="Window Menu">Window Menu</a>'s Info dialog. A '\*' at the beginning and/or at the end can be used for partial string matching (e.g. "\*MyClass\*" will match any window class containing the string "MyClass"). Leave is field empty if the window rule is not to match on the window's class.
- *Window Name*: Defines the title of the window, note that this must be correctly set for when the window is first created (or when the window rules are reapplied). A '\*' at the beginning and/or at the end can be used for partial string matching (e.g. "\*Mozilla Firefox" will match any window whose title ends in the string "Mozilla Firefox"). Leave is field empty if the window rule is not to match on the window's title.
- **Program Name**: Defines the full path of the program which created the window, this can be found in the Window Menu's Info dialog. A '\*' at the beginning and/or at the end can be used for partial string matching (e.g. "\*\firefox.exe" will match any window created by Firefox). Leave is field empty if the window rule is not to match on the window's executable.

**Note:** The Program Name is not support on early versions of Windows, e.g. Win9x.

• *Window rule enabled*: If not checked the window rule is simply ignored, this option can be useful when tracking down problems.

**Note:** When using the <u>Window Menu</u>'s Add Window Rule feature the three name fields are set to the current window's details, it is very unlikely that all three fields are required for the identification, particularly the Window Name.

## **Settings**

**Note:** It is strongly advised that minimal changes made to avoid unwanted side effects; VirtuaWin handles most windows correctly so only change settings which directly address a problem.

• Automatically close windows of this type: If checked any window of this type will be automatically closed as soon as they are detected.

**Note:** This option is applied regardless of whether VirtuaWin is configured to manage the window - see option <u>Manage windows of this type</u>.

• *Make windows of this type always on top*: If checked any window of this type will have their Window's 'Always on top' property set which keeps the window above other windows.

**Note:** This option is applied regardless of whether VirtuaWin is configured to manage the window - see option <u>Manage windows of this type</u>.

- *Manage windows of this type*: Three radio buttons give the options to Never manage, Always manage or Let VirtuaWin decide (default).
- *Main window of multiple taskbar buttons*: If checked the window is flagged as the main visible window for an application that uses multiple taskbar buttons to access this window. For example Excel creates a taskbar button for each file opened but has only one main window, clicking on each taskbar button simply changes which file is viewed in the main window. This option enables VirtuaWin to identify these main windows and handle them properly by removing the additional taskbar button for the main window (this button is typically removed by the application). Only use this option if changing desktops leads to the creation of an additional taskbar button.
- *Show windows of this type on all desktops*: If checked any window of this type will be automatically shown on all desktops. This differs from the window not being managed in several ways, firstly the window will not appear in the main Window List menu, also when changing desktops VirtuaWin will not give the focus to a non-managed window.
- *Keep all windows of this application together*: If checked all windows created by the window's application are kept together, i.e. VirtuaWin will move all the

application's windows when one of them is moved and if one is flagged as Always Show then all the windows will be.

- Automatically move windows of this type to Desktop: If checked the window will be automatically moved to the given desktop. VirtuaWin will move the window immediately on start-up or when window rules are reapplied, when VirtuaWin is already running and a new window is opened it is either moved immediately if *Move windows immediately* is enabled or when the window is first dismissed (either by using the user dismissing the window or by changing desktops).
- *On hidden window activation*: This is a window specific setting for the main Setup On hidden window activation option, see this option for more information. When this option is set to **Default configured in main Setup** the main Setup option's setting is used.
- *Method for hiding the window*: Specifies what method should be used to hide windows of this type, the standard method works well for most windows and is the most efficient, however some windows (most notably Explorer, IExplorer and some .NET dialogs) mysteriously disappear when this method is used. To avoid this problem these windows must be hidden using the move window method. The minimizing window method is the most efficient to use when the taskbar button is to be kept shown.

**Note:** The Windows 'Animate windows when minimizing and maximizing' performance option greatly affects the minimizing hiding method; if you need to use this method of hiding it is recommended that you disable this option.

• *Method for hiding taskbar button*: Specifies what method, if any, is used to hide the window's taskbar button, the standard method is the most efficient but can only be used when the window is hidden using the standard method. If set to *Show*, VirtuaWin will not hide the taskbar button, this is useful when the application uses the taskbar button to show events (such as a new message etc), if the taskbar button is hidden the event could be missed - only *move window* or *minimizing window* hiding methods can be used with this setting. The *Hide by using toolwin flag* is an alternative hiding method which must be used when the window's hiding method is not standard method.

# **Hidden Configuration Options**

Some of the more obscure VirtuaWin configuration options have not been added to the Setup dialog as this dialog is already rather complex. Following is a list of this hidden configurations, if you wish to change a setting please refer to the <a href="Editing a Hidden Option"><u>Editing a Hidden Option</u></a> section below.

| Option name     | Values | Description   |
|-----------------|--------|---|
| initialDesktop  | #      | Sets the initial (or startup) desktop of VirtuaWin to #.  |
| lastDeskNoDelay | 0 or 1 | Disables the one second delay before updating the last desktop as used by the <u>NAV: Return to last</u> <u>desktop</u> . |

## **Editing a Hidden Option**

To change the value of a hidden configuration option you need to manually edit the <u>virtuawin.cfg</u> file. To do this:

- 1. Ensures your virtuawin.cfg is the latest version as earlier versions may not have the option you are trying to set. To do this:
- a. Open the VirtuaWin Setup Dialog.
- Make any change so that the **Apply** button becomes active and press it.
- Undo your change and press the **Apply** button again.
- In the Setup Dialog got to the Expert tab and click on **Explore Config**, then click on **Ok** to close the setup dialog.
- Exit VirtuaWin.

- In the Explorer window opened in step 2 locate the virtuawin.cfg file and open this in a text editor (such as notepad).
- Locate the line which starts '<option-name># ', this will typically be near the bottom of the file. If the option cannot be found you most likely using an earlier version of VirtuaWin.
- Change the text to the right of the '<option-name># ' marker to your required new value and save the file.
- Restart VirtuaWin, your change should now take effect.

## WinList Module

The WinList module is used for recovering any window that might have been lost by VirtuaWin.

To open WinList's recovery dialog open VirtuaWin's <u>Setup Dialog</u>, go to the Modules tab and select WinList in the Module List, then press the configure button. This will open the WinList dialog listing all the windows that currently exists. Note that there are many system windows that are hidden by default, which are also shown in this list. It is possible to undo the latest action if the window list dialog is not closed.

The Flag column (F) indicates the current handling of the window by VirtuaWin, it will have one of the following values:

- 1 .. 20: Known managed window, the number indicates the desktop it belongs to.
- U: Known unmanaged window.
- **V**: Unknown visible window (rare).
- **H**: Unknown hidden window (possibly lost).

The Status column (S) indicates the current status of the window according to VirtuaWin, this column is only set when VirtuaWin is managing the window (i.e. the F column is set to a desktop number) in which case it will have one of the following values:

- **O**: Okay the window is responding normally.
- **E**: Elevated the window is considered elevated (Vista or Win7), VirtuaWin cannot manage these windows as it does not have the require permissions to hide them.
- **H**: Hung the window has stopped responding, VirtuaWin will continue to manage and monitor the window, it will be recovered as soon as it starts responding. The VirtuaWin system try icon should be flashing to warn you of

this issue.

The window's application classname and title bar text are displayed in the *Class* and *Title* columns, if the window does not have a title "<None>" will be displayed instead.

#### **Custom Icons**

It's possible to use custom icons on the system tray. For using this feature you need to get a tool for creating the icons (or download custom icons for the website – see below) and add them to the "icons" directory in the install directory of VirtuaWin.

VirtuaWin uses icons 0.ico, 1.ico, 2.ico ... n.ico where 'n' is the number of desktops used. The 0.ico is used when the user disables VirtuaWin; the others are used when on the respective desktop (i.e. 3.ico is used when on desktop 3). If any icon is missing from the icons directory VirtuaWin will use a default icon; this will be the main VirtuaWin icon for desktops greater than 9.

When creating you own, the icons must have a size of 16x16 pixels and the color palette must be restricted to the 16 'dos' colors for pre-XP versions of windows. The naming of the icon files must be as described above.

VirtuaWin associates the icons with desktops from left to right row by row, meaning that if you have a 3 by 3-desktop area the icon positions will be:

123

456

789

And a 2 by 3 will be:

12

34

56

If there are custom icons present, they will be used in both 2 by 2 and all other desktop sizes. Check <a href="http://virtuawin.sourceforge.net/">http://virtuawin.sourceforge.net/</a> for other icons, and if you make your own icon set please make them available to all by submitting them as a <a href="SourceForge patch">SourceForge patch</a>.

**Please Note:** For copyright reasons we require your submission to include some method of contact (e.g. SourceForge user id) and confirmation that the icons are free of copyright. If these are not given we will not be able to publish them on

our website. The submission is open so please obfuscate any email address.

## **External Interface**

Most VirtuaWin commands and features can be accessed by sending windows messages to the VirtuaWinMainClass window, this is also known as the Module interface. Any external program can locate this VirtuaWin window and send it commands, or use VirtuaWin's **-msg** command-line option, the format of this option is as follows:

```
VirtuaWin.exe -msg <Msg> <wParam> <lParam>
```

Where <Msg>, <wParam> and <1Param> are numbers as per the standard Windows SendMessage function. A complete and up-to-date list of supported messages can be found in Messages.h in the SDK package, the following list may not be fully up-to-date:

| Msg  | wParam | lParam | Description                          |
|------|--------|--------|--------------------------------------|
| 1034 | #      | -      | Change to desktop #.                 |
| 1034 | 1025   | _      | Change to previous desktop.          |
| 1034 | 1026   | -      | Change to next desktop.              |
| 1034 | 1035   | -      | Change to next desktop to the left.  |
| 1034 | 1036   | -      | Change to next desktop to the right. |
| 1034 | 1037   | -      | Change to next desktop above.        |
| 1034 | 1038   | _      | Change to next desktop below.        |

| 1039 | -            | _    | Exit VirtuaWin.   |
|------|--------------|------|---|
| 1040 | -            | _    | Open Setup dialog.  |
| 1041 | -            | _    | Remove VirtuaWin's systray icon.                                      |
| 1042 | -            | -    | Show VirtuaWin's systray icon.  |
| 1043 | -            | -    | Open VirtuaWin's on-line help.  |
| 1044 | -            | -    | Gather all windows onto the current desktop.                          |
| 1045 | -            | -    | Returns the width of the current layout, i.e. how many desktops wide. |
| 1046 | -            | -    | Returns the depth of the current layout, i.e. how many desktops deep. |
| 1048 | -            | -    | Returns the current desktop number.                                   |
| 1049 | hWnd or<br>0 | #    | Move window to desktop #.   |
| 1049 | hWnd or<br>0 | 1025 | Move window to previous desktop.                                      |
|      |              |      |   |

| 1049 | hWnd or<br>0 | 1026              | Move window to next desktop.   |
|------|--------------|-------------------|--|
| 1049 | hWnd or<br>0 | 1035              | Move window to next desktop to the left.   |
| 1049 | hWnd or<br>0 | 1036              | Move window to next desktop to the right.  |
| 1049 | hWnd or<br>0 | 1037              | Move window to next desktop above.   |
| 1049 | hWnd or<br>0 | 1038              | Move window to next desktop below.   |
| 1049 | hWnd or<br>0 | -#,<br>-1025,<br> | As with positive lParam values only VirtuaWin changes desktops as well, i.e. moves window and follows. |
| 1050 | hWnd or<br>0 | -1                | Toggle whether the window is shown on all desktops.  |
| 1050 | hWnd or<br>0 | 0                 | Do not show window on all desktops.  |
| 1050 | hWnd or<br>0 | 1                 | Show window on all desktops.   |
| 1051 | hWnd         | 0                 | Give the window the focus, i.e. make it the foreground window.   |

|      | <u> </u>                      |   |
|------|-------------------------------|---|
| 0    | 0                             | Get VirtuaWin to select the best window to give the current focus to.   |
| hWnd | #                             | Give the window the focus for the given desktop, i.e. sets the z-order so the window will be given the focus when next on that desktop. |
| hWnd | -                             | Send the VirtuaWin installation path to hwnd via a WM_COPYDATA message.   |
| hWnd | -                             | Send the VirtuaWin user configuration path to hwnd via a WM_COPYDATA message.   |
| hWnd | 0                             | Access window - use method defined by 'On hidden window activation' setting.  |
| hWnd | 1                             | Access window - move to the current desktop.  |
| hWnd | 2                             | Access window - shown on the current desktop.   |
| hWnd | 3                             | Access window - change to the window's desktop.   |
| hWnd | -1                            | Use window's 'On hidden window activation' setting (if set to ignore nothing will happen).  |
| hWnd | -                             | Returns information about the window, see<br>Messages.h for more information.   |
|      | hWnd hWnd hWnd hWnd hWnd hWnd | hWnd # hWnd - hWnd 0 hWnd 1 hWnd 2 hWnd 3 hWnd 3  |

| 1065 | ?            | ? | Controls desktop thumbnail image generation, see<br>Messages.h for more information.   |
|------|--------------|---|--|
| 1066 | 0            | - | Returns the current VirtuaWin running state (0 = disabled, 1 = enabled).   |
| 1066 | 1            | - | Toggles the current VirtuaWin running state, returns the previous state.   |
| 1066 | 2            | - | Disables VirtuaWin, returns the previous running state.  |
| 1066 | 3            | - | Enables VirtuaWin, returns the previous running state.   |
| 1067 | hWnd         | # | Send the name of desktop # (or current desktop if # = 0) to hwnd via a WM_COPYDATA message.  |
| 1068 | -            | - | Returns the largest possible desktop number.   |
| 1069 | hWnd or<br>0 | 0 | Force VirtuaWin to NOT manage the given window. <b>Note</b> : VirtuaWin will remember this setting for the lifespan of the window, only a 'manage window' message (see below) can change this. |
| 1069 | hWnd or<br>0 | 1 | Force VirtuaWin to manage the given window.  |
|      |              |   |  |

| 1070 | cmd-id | ?    | Execute hotkey command <i>cmd-id</i> (see second column in vwCommands.def) as if the user pressed the hotkey. The lParam value is split into to parts, the lower 16 bits (LOWORD(lParam)) is the desktop number (if required but the hotkey command) and the higher 16 bits (HIWORD(lParam)) is the modifier, only bit vwHOTKEY_WIN_MOUSE is used. |
|------|--------|------|--|
| 1071 | hWnd   | 0    | Remove all items added to the control menu, use WM_COPYDATA to add items, see feature request 2980468 for more information and examples.   |
| 1071 | hWnd   | #    | Remove added item # from the control menu.   |
| 1072 | 0      | -    | Get the handle of the current module handling desktop changes.   |
| 1072 | 1      | hWnd | Set the handle of the module handling desktop changes, returns 2 if successful, 1 otherwise.   |
| 1072 | 2      | hWnd | Unsets the current desktop change handling module.   |
| 1072 | 3      | -    | Execute the desktop change - only sent but the module controlling desktop changing. See Messages.h for more information.   |
| 1073 | -      | -    | Re-initialize all windows visible on the current desktop, applying all window rules appropriately. Window settings like 'Always show' will be reset.   |

| 1074 | - | Return current Inervt Up/Down setting, Re-<br>initialize all windows visible on the current<br>desktop, applying all window rules appropriately.<br>Window settings like 'Always show' will be reset. |
|------|---|---|

#### Where:

- means not used (use a 0).

hWnd Numeric handle to the window required.

hWnd Numeric handle to the window required or 0 for the window which or 0 currently has the focus.

# **Locking or Hiding Desktops**

As applications run independently of VirtuaWin, a 'locked desktop' concept is flawed and cannot be made secure. For example, an unwanted guest could use the task manager to suspend the VirtuaWin process and then use any window restore tool (like WinList) to access the hidden windows. From a security perspective, the only thing a locked desktop achieves is to prove you have something to hide!

A more suitable approach would be 'stealth', what is not seen is unlikely to be found. A hidden desktop is no more secure than a locked desktop when someone else knows it exists, so don't tell anyone!

VirtuaWin can be configured to support 1 or more hidden desktops by assigning a desktop access hotkey to a desktop that is outside of the main Desktop Layout size. For example, if a 2 by 2 layout is used, assigning a hotkey to desktop 8 will give you direct access to this desktop yet the desktop and its contents will be hidden in every other aspect.

To configure a hidden desktop, use the main VirtuaWin Setup to change your Desktop Layout so that your required hidden desktop becomes part of the layout, e.g. to enable desktop 8 change to a 4 by 2 layout. Change to this new desktop and use Setup to assign a hotkey, make sure the hotkey is not something you could accidentally type and don't forget it! Last, restore the desktop layout so that the hidden desktop is outside this range (i.e. back to a 2 by 2 layout) and check you can still access the hidden desktop using your assigned hotkey.

As a hidden desktop will not appear on any VirtuaWin menu, moving a window to it must be done by either making the window sticky or by going to the desktop and using the Window List menu to Assign the window.

Hiding desktops in VirtuaWin is deliberately low profile, the less this feature is known the more secure it is.

# **Configuration Files**

With the exception of **userpath.cfg** all configuration files are located in the user's 'Application Data/VirtuaWin' directory, use the Setup Expert *Explore Config* button to locate this directory. When a user runs VirtuaWin for the first time all configuration files located in the VirtuaWin installation directory are copied into the user's own VirtuaWin directory within their Application Data area. The following configuration files are used by VirtuaWin:

## userpath.cfg

If present in the installation directory it contains a single line defining the directory containing the user's configuration files. This file can be used to solve problems caused by user permissions and roaming profiles. The string within the file can user environment variables by using the format '\${<variable-name>}' (e.g. \${HOMEPATH}) and can use the built in variable \${VIRTUAWIN\_PATH} to use the VirtuaWin installation directory. For example, the following could be used if a fixed path with the same setup for all users was required:

```
c:\VirtuaWin_config\
```

The following uses a fixed path but allows different configurations per user:

```
c:\VirtuaWin_conf\${USERNAME}
```

Note that VirtuaWin will create the directory if it does not already exist.

## virtuawin.cfg

Main user configuration file, the Setup Dialog settings are stored in this file.

## window.cfg

Configuration file used to store the specific Window Rule setup.

## module.cfg

Contains the list of currently disabled modules.