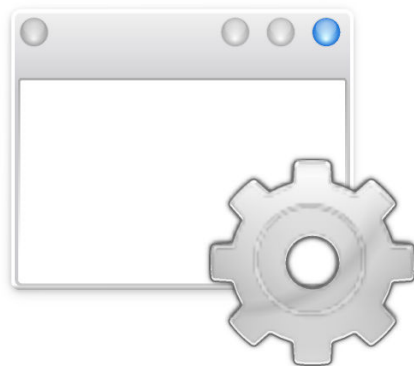


# Window Behavior

Mike McBride  
Jost Schenck



## Window Behavior

## Contents

<b>1 Window Behavior</b>	<b>4</b>
1.1 Focus	4
1.1.1 Focus Policy	4
1.1.2 Focus stealing prevention level	5
1.1.3 Raising window	5
1.2 Titlebar Actions	6
1.2.1 Titlebar double-click	6
1.2.2 <b>Titlebar &amp; Frame</b>	6
1.2.3 <b>Maximize Button</b>	7
1.3 Window Actions	7
1.3.1 <b>Inactive Inner Window</b>	7
1.3.2 <b>Inner Window, Titlebar &amp; Frame</b>	7
1.4 Moving	8
1.4.1 <b>Windows</b>	8
1.4.2 <b>Snap Zones</b>	8
1.5 Advanced	8

# 1 Window Behavior

In the upper part of this control module you can see several tabs: **Focus**, **Titlebar Actions**, **Window Actions**, **Moving** and **Advanced**. In the **Focus** panel you can configure how windows gain or lose focus, i.e. become active or inactive. Using **Titlebar Actions** and **Window Actions** you can configure how titlebars and windows react to mouse clicks. **Moving** allows you to configure how windows move and place themselves when started. The **Advanced** options cover some specialized options like ‘window tabbing’ and ‘window shading’.

## NOTE

Please note that the configuration in this module will not take effect if you do not use KDE’s native window manager, KWin. If you do use a different window manager, please refer to its documentation for how to customize window behavior.

## 1.1 Focus

The ‘focus’ of the desktop refers to the window which the user is currently working on. The window with focus is often referred to as the ‘active window’.

Focus does not necessarily mean the window is the one at the front — this is referred to as ‘raised’, and although this is configured here as well, focus and raising of windows are configured independently.

### 1.1.1 Focus Policy

There are six methods KDE can use to determine the current focus:

#### Click To Focus

A window becomes active when you click into it. This behaviour is common on other operating systems and likely what you want.

#### Click To Focus - Mouse Precedence

This is mostly the same as **Click To Focus**. If an active window has to be chosen by the system (e.g. because the currently active one was closed) the window under the mouse is the preferred candidate. Unusual, but possible variant of **Click To Focus**.

#### Focus Follows Mouse

Moving the mouse pointer actively over a normal window activates it. New windows such as the mini command line invoked with **Alt+F2** will receive the focus, without you having to point the mouse at them explicitly. e.g. windows randomly appearing under the mouse will not gain the focus. Focus stealing prevention takes place as usual. Think as **Click To Focus** just without having to actually click.

In other window managers, this is sometimes known as ‘Sloppy focus follows mouse’.

#### Focus Follows Mouse - Mouse Precedence

This is mostly the same as **Focus Follows Mouse**. If an active window has to be chosen by the system (e.g. because the currently active one was closed) the window under the mouse is the preferred candidate. Choose this, if you want a hover controlled focus.

#### Focus Under Mouse

The window that happens to be under the mouse pointer becomes active. If the mouse is not over a window (for instance, it’s on the desktop) the last window that was under the mouse has focus. New windows such as the mini command line invoked with **Alt+F2** will not receive the focus, you must move the mouse over them to type.

### Focus Strictly Under Mouse

Similar to **Focus Under Mouse**, but even more strict with its interpretation. Only the window under the mouse pointer is active. If the mouse pointer is not over a window, no window has focus. New windows such as the mini command line invoked with **Alt+F2** will not receive the focus, you must move the mouse over them to type.

#### NOTE

Note that **Focus Under Mouse** and **Focus Strictly Under Mouse** prevent certain features, such as **Focus stealing prevention** and the **Alt+Tab** walk-through-windows dialog, from working properly.

### 1.1.2 Focus stealing prevention level

This option specifies how much KWin will try to prevent unwanted focus stealing caused by unexpected activation of new windows.

#### None

Prevention is turned off and new windows always become activated.

#### Low

Prevention is enabled; when some window does not have support for the underlying mechanism and KWin cannot reliably decide whether to activate the window or not, it will be activated. This setting may have both worse and better results than the medium level, depending on the applications.

#### Medium

Prevention is enabled.

#### High

New windows get activated only if no window is currently active or if they belong to the currently active application. This setting is probably not really usable when not using mouse focus policy.

#### Extreme

All windows must be explicitly activated by the user.

Windows that are prevented from stealing focus are marked as demanding attention, which by default means their taskbar entry will be highlighted. This can be changed in the Notifications control module.

### 1.1.3 Raising window

Once you have determined the focus policy, there are the window raising options.

With a click to focus policy by default **Click raises active window** is enabled and raise on hover is not available.

With a hover to focus policy you can alternatively use auto raise. By placing a mark in front of **Raise on hover, delayed by**, KDE can bring a window to the front if the mouse is over that window for a specified period of time. You can determine the delay for this option by using the spin box control.

## Window Behavior

### TIP

Setting the delay too short will cause a rapid fire changing of windows, which can be quite distracting. Most people will like a delay of 100-300 ms. This is responsive, but it will let you slide over the corners of a window on your way to your destination without bringing that window to the front.

If you do not use auto raise, make sure the **Click raises active window** option has a mark in front of it. You will not be happy with both auto raise and **Click raise active window** disabled, the net effect is that windows are not raised at all.

## 1.2 Titlebar Actions

In this panel you can configure what happens to windows when a mousebutton is clicked on their titlebars.

### 1.2.1 Titlebar double-click

In this drop down box you can select either **Shade**, several variations of **Maximize** or **Lower, Close** and **On All Desktops**.

Selecting **Maximize** causes KDE to maximize the window whenever you doubleclick on the titlebar. You can further choose to maximize windows only horizontally or only vertically.

**Shade**, on the other hand, causes the window to be reduced to simply the titlebar. Double clicking on the titlebar again, restores the window to its normal size.

Similar options are available for **Wheel event**.

### TIP

You can have windows automatically unshade when you simply place the mouse over their shaded titlebar. Just check the **Enable hover** check box in the **Advanced** tab of this module. This is a great way to reclaim desktop space when you are cutting and pasting between a lot of windows, for example.

### 1.2.2 Titlebar & Frame

This section allows you to determine what happens when you single click on the titlebar or frame of a window. Notice that you can have different actions associated with the same click depending on whether the window is active or not.

For each combination of mousebuttons, Active and Inactive, you can select the most appropriate choice. The actions are as follows:

#### Raise

Will bring the window to the top of the display. All other windows which overlap with this one, will be hidden 'below' it.

#### Lower

Will move this window to the bottom of the display. This will get the window out of the way.

#### Toggle Raise & Lower

This will raise windows which are not on top, and lower windows which are already on top.

#### Nothing

Just like it says. Nothing happens.

## Window Behavior

### Operations Menu

Will bring up a small submenu, where you can choose window related commands (i.e. Maximize, Minimize, Close, etc.).

### 1.2.3 Maximize Button

This section allows you to determine the behavior of the three mouse buttons onto the maximize button. You have the choice between vertical only, horizontal only or both directions.

## 1.3 Window Actions

### 1.3.1 Inactive Inner Window

This part of the module, allows you to configure what happens when you click on an inactive window, with any of the three mouse buttons or use the mouse wheel.

Your choices are as follows:

#### Activate, Raise & Pass Click

This makes the clicked window active, raises it to the top of the display, and passes a mouse click to the application within the window.

#### Activate & Pass Click

This makes the clicked window active and passes a mouse click to the application within the window.

#### Activate

This simply makes the clicked window active. The mouse click is not passed on to the application within the window.

#### Activate & Raise

This makes the clicked window active and raises the window to the top of the display. The mouse click is not passed on to the application within the window.

### 1.3.2 Inner Window, Titlebar & Frame

This bottom section, allows you to configure additional actions, when a modifier key (by default **Alt**) is pressed, and a mouse click is made on a window.

Once again, you can select different actions for Left, Middle and Right button clicks and the **Mouse wheel**.

Your choices are:

#### Move

Allows you to drag the selected window around the desktop.

#### Lower

Will move this window to the bottom of the display. This will get the window out of the way.

#### Nothing

Just like it says. Nothing happens.

#### Raise

Will bring the window to the top of the display. All other windows which overlap with this one, will be hidden 'below' it.

## Window Behavior

### **Resize**

Allows you to change the size of the selected window.

### **Toggle Raise & Lower**

This will raise windows which are not on top, and lower windows which are already on top.

### **Activate**

Make this window active.

## **1.4 Moving**

### **1.4.1 Windows**

The options here determine how windows appear on screen when you are moving them.

#### **Display window geometry when moving or resizing**

Enable this option if you want a window's geometry to be displayed while it is being moved or resized. The window position relative to the top-left corner of the screen is displayed together with its size.

### **1.4.2 Snap Zones**

The rest of this page allows you to configure the **Snap Zones**. These are like a magnetic field along the side of the desktop and each window, which will make windows snap alongside when moved near.

#### **Border snap zone:**

Here you can set the snap zone for screen borders. Moving a window within the configured distance will make it snap to the edge of the desktop.

#### **Window snap zone:**

Here you can set the snap zone for windows. As with screen borders, moving a window near to another will make it snap to the edge as if the windows were magnetized.

#### **Center snap zone:**

Here you can set the snap zone for the screen center, i.e. the 'strength' of the magnetic field which will make windows snap to the center of the screen when moved near it.

#### **Snap windows only when overlapping**

If checked, windows will not snap together if they are only near each other, they must be overlapping, by the configured amount or less.

## **1.5 Advanced**

In the **Advanced** panel you can do more advanced fine tuning to the window behavior.

### **SHADING**

#### **Enable hover**

If this option is enabled, a shaded window will un-shade automatically when the mouse pointer has been over the titlebar for some time. Use the spinbox to configure the delay un-shading.



## Window Behavior

### WINDOW TABBING

#### **Automatically group similar windows**

When turned on attempt to automatically detect when a newly opened window is related to an existing one and place them in the same window group.

#### **Switch to automatically grouped windows immediately**

When turned on immediately switch to any new window tabs that were automatically added to the current group.

#### **Placement**

The placement policy determines where a new window will appear on the desktop. **Smart** will try to achieve a minimum overlap of windows, **Cascade** will cascade the windows, and **Random** will use a random position. **Centered** will open all new windows in the center of the screen, and **Zero-Cornered** will open all windows with their top left corner in the top left corner of the screen.

### SPECIAL WINDOW

#### **Hide utility windows for inactive applications**

When turned on, utility windows (tool windows, torn-off menus,...) of inactive applications will be hidden and will be shown only when the application becomes active. Note that applications have to mark the windows with the proper window type for this feature to work.