

SmartButtons 1.3 © www.LogicZone.de

What is a SmartButton:

A SmartButton is an image button based on a panel used in MaxGui and uses a BlitzMax module.

What is so smart about it:

While a panel can only serve events in its rectangular shape, the SmartButton is far more flexible. The user can define *one or more* hotspot regions where the SmartButton should generate events.

What is a SmartButton hotspot region:

Hotspot regions can consist out of rectangles, circles, lines, ovals and polygons (any shape)! That means, you don't need a panel for every event you want to catch, simply define more than one region. Every hotspot region produces an event when a region is entered (hover), left mouse button up and down, right mouse button up and down, mouse wheel up and down and you can run your own coded routines. Having said that, you don't have to! As the SmartButtons supporting also certain actions. Additionally you can use a timertick with interval as trigger and special regions allows you to customize your SmartButton further:

Enter SmartButton	all actions in this region are run when the SmartButton itself is entered. Useful when the SmartButton rectangle is suffice as hotspot.
Leave SmartButton	all actions in this region are run when the SmartButton is left. Good to use to reset your graphics/SmartButton.
SmartButton creation	all actions are run directly after the SmartButton is created. Perfect when you need to initialize something or to start a timer.
SmartButton Timer	all actions which should run at certain intervals. Run actions at different times without the need of a manually trigger.

What is a SmartButton action:

An Action can be SetPicture, PlaySound, Simple Animation...

Every region can hold *one or more* actions. The trigger decides when to act. These actions once created are completely independent. They will run every time the event is fired. All SmartButton actions and events are hook based and will occur without delay.

How to install the SmartButton module:

Unzip the SmartButton archive and put it in your BlitzMax mod folder. The structure should look at the end like this: `..\BlitzMax\mod\logiczone.mod\smartbuttons.mod\.`

Run `..\BlitzMax\bin\docmods.exe` to update your BlitzMax documentation (simply double click).

After that the new commands from the SmartButton module are highlighted and the F1 help is available on your system. The documentation can be found under 'Third Party Modules' in the Help section of the IDE. As all third party module are not automatically imported, you should do it manually when needed: `Import logiczone.SmartButtons`

Definition of a SmartButton:

To hold all the information needed for a SmartButton a xml config file is used. You can find all created regions and defined actions with all paths to external media files in here. If you feed the SmartButton module only by your own code, this xml config file is not needed at all. But when you want to use this SmartButton in Logic Gui, this is the way to store the information. While it is not easy to build such a xml config file manually a SmartButton Editor will do it for you automatically.

The SmartButton Editor:

This editor let's you create visual the needed hotspot regions.

Adds actions to your regions.

Is able to store or load a SmartButton config file (*.smb).

The SmartButton Editor

Workflow:

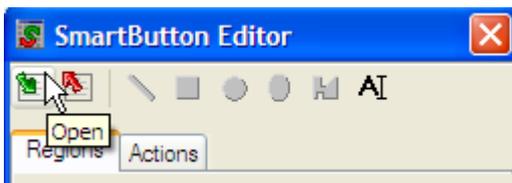
1. Open picture as image button background
2. Define regions where events should be generated
3. Select a certain region and add an action if needed
4. Save SmartButton config file

or

1. Load SmartButton config file (loads background and shows regions)
2. Add or delete regions
3. Add, delete or edit actions
4. Save SmartButton config file

Creating a SmartButton:

1. Open picture as image button background



A file open requester pops up.

You can choose SmartButton (default) or Image file as filetype. Change to image file and open your background picture for the image button.

Or drag and drop an image file onto the SmartButton Editor Canvas as main picture, or you can drop also

the .smb SmartButton config file, which will then load the complete definition.

2. Define regions where events should be generated

Add a certain region by selecting the corresponding icon from the iconstrip. The cursor will change to a cross and you can start defining the region. Click on the first point on the picture where the region should be located. You can release the mouse and move around to check how the region looks like until you find the correct size and position, then click again to add that region to the list.



In the example above you can see the rectangle and circle region. The region shows up transparent in red while defining and in yellow when already added to the list.

Define as many regions as needed for your SmartButton:

Stop creating new regions by a right click! (or press ESC)

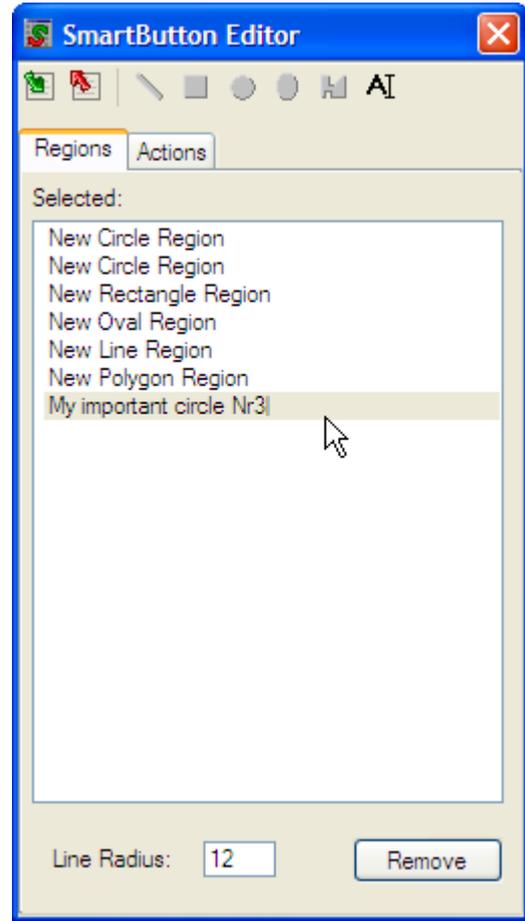
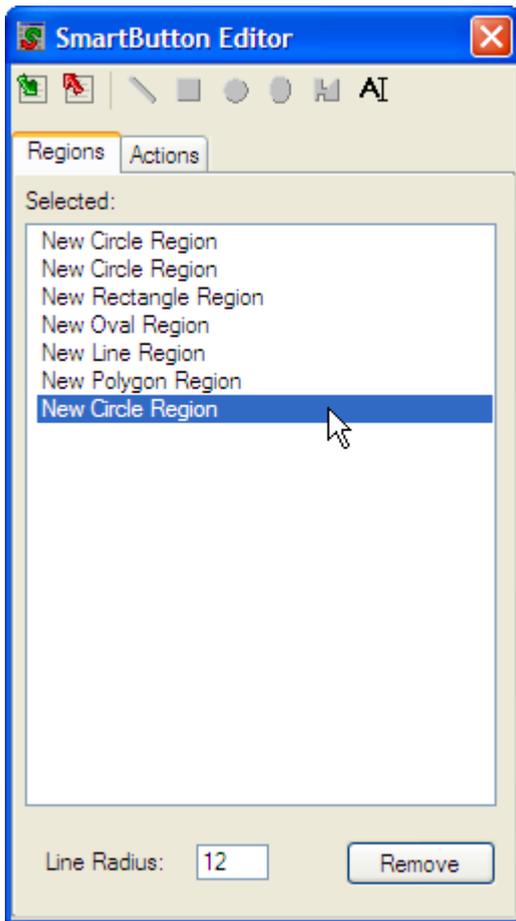


When using a line as region you can type a specified thickness in the Line Radius TextField, or just simply move the mouse wheel!

You can select a region and move it when you hold the mouse button down. Thus you are able to adjust the hotspot region afterwards.

Multi selection of regions is also possible. Move all added regions at once if you want. Shift click the region you want to add/subtract from the multi selection. Single click on a region deselects all other. A click on the background (no region) deselects all.

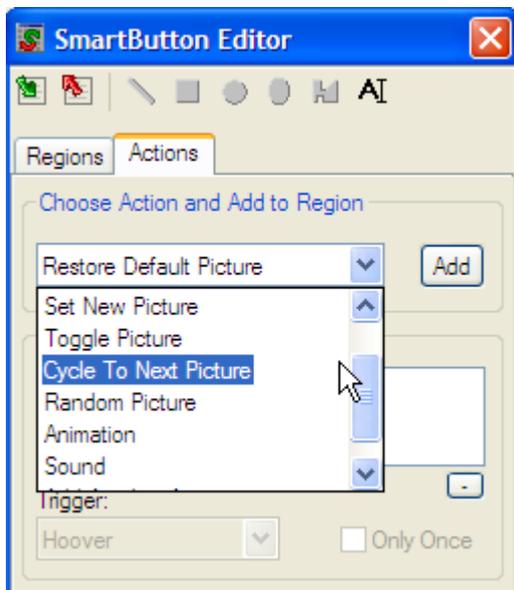
Every created region will appear in the region ListBox with a generic name:



Right click on a region name in the ListBox to give it another name for better identification. A double click on a certain region in the ListBox switches automatically to the action tab and shows the corresponding actions.

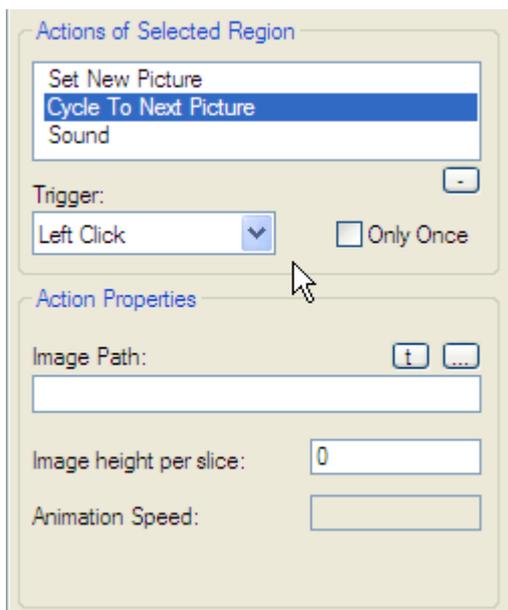
3. Select a certain region and add an action if needed

When you select a region in the SmartButton, the corresponding region in the ListBox is highlighted and vice versa. With a region selected you can click on the 'Action' tab:



A ComboBox shows a list of available actions. Choose one and click on 'Add'. You can add more than one action to a region, it should just make sense to you... Perhaps one action for a left click and one action on right click. Or maybe two actions for hover trigger, e.g. a new picture and a suitable sound.

Click on an action in the ListBox to define the action parameter. Every action has at least a 'Trigger' and you can select if this action should run every time the corresponding region event occurs or **only once**.

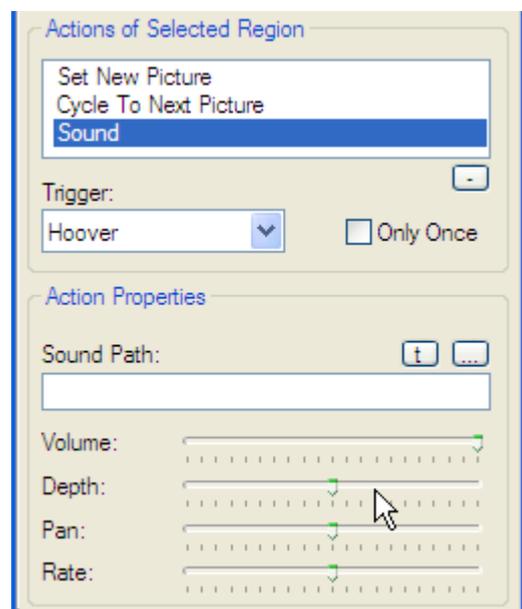


When a new picture is needed you can define the path to it in the Action Properties. Type it in or use the ... three dots to open a requester. Test the new picture by pressing the t (testbutton) which show the image temporarily until you proceed editing.

When more than one picture is needed a pixmap like an iconstrip is used. That means all single frames are together in one image. Only difference is may that all images are aligned in a vertical row. In the 'Image height per slice' field you can define the size of a single frame.

If an animation is used you also need to fill in the 'Animation Speed' which is defined in frames per second.

If you used a sound action you need to set the path to the sound file and the Volume, Depth, Pan and Rate. Press the t (testbutton) to check how it sounds when playing with your settings.



5. Save SmartButton config file

When all settings are done you can save the SmartButton config file.

You don't need to apply any parameter, everything is done on the fly!

If something went wrong, either edit it or delete and re-create it, all adjustments are quite quick to do, nothing really important can be lost.

Click on the red arrow in the iconstrip to open a save requester, give your SmartButton a meaningful name and click ok.

All paths from external media are converted to relative paths where possible and written to the xml .smb config file. You can organize your SmartButton and it's media in a certain folder, or the media may in a descriptive subfolder. Thus the SmartButton can be copied more easily and independetly.

Finished!

Extra Information

Two special regions appear in the SmartButton Editor when you start up, 'Enter SmartButton' and 'Leave SmartButton'. These two regions represent the SmartButton itself! When you enter the SmartButton all actions are run defined in the Enter SmartButton region (Trigger should be Hover). When you leave the SmartButton all actions are run in the defined Leave SmartButton region. Those regions are independent from the hotspot regions you define. As both regions are the SmartButton itself and the SmartButton can be resized by it's SetGadgetLayout command, those regions are dynamic! Another advantage is that you could for example reset the SmartButton to it's original state when the user leaves the SmartButton area (loses focus), autonomous from any other hotspot region.

The SmartButton module handles also the mouse wheel events. Actions can be triggered on wheel up and wheel down events. The module generates an EVENT_GADGETOPEN for every wheel up and an EVENT_GADGETCLOSE for every wheel down event. The data field contains the delta clicks and the extra field the regions extra object.

Trigger: Left Click Up and Down, Right Click Up and Down, Hover, Wheel Up and Wheel Down, Timer Tick

Actions: restore default picture, make picture grey, set new picture, toggle picture, cycle to next picture, random picture, play animation, play sound, show a tip, run actions, start SmartButton timer, stop SmartButton timer, execute function, blur picture, invert picture, highlight picture, color picture, set spinner picture, spinner forward, spinner backward

ShowTip:

A ShowTip Action is now available in the SmartButton module. You can enter a text which should be displayed and a time in seconds how long. The tip will show up after some time of no mouse movement on a certain region (use only your own defined hotspot regions not the Enter or Leave Region). The delay can be set in seconds as well. When the tip is displayed and you move the mouse the tip will follow. The time how long the tip is shown starts running after the mouse stands still. When you leave the region or the SmartButton the tip is removed. If the tip should only be displayed the first time the user enters a region, switch the 'Only Once' flag on. To have the tip outside the region which may show some important things you can define also an offset in pixel. The offset is always added to the mouse pointer coordinates, you can use negative and positive values to reach any position. The text of the tip is put in a MaxGui label (centered by default) where you can define also the width and height, which is then responsible for the word wrap, just use what fit your needs.

SmartButton Timer and Timer Ticks:

A trigger 'Timer_Tick' is a trigger, which can automatically trigger your action without the need to enter a specific region. Every action is now able to hold an interval value which is automatically set from the used trigger. The trigger integer is split into two sections, the lower two bytes hold the bits when an action is triggered and the upper two bytes can hold an interval value. Two new actions, StartTimer and StopTimer, are now available to steer the behavior. The create StartTimer action can set the 'Hertz' value for the timer and determines the speed of the timerticks. The StartTimer and StopTimer, starts and stops the timer respectively. A new TimerRegion holds all actions which should run triggered by a timertick. Add all actions you require to this region and give every action an interval (same or different). When a timertick occurs the module decrements the interval value of every action in the TimerRegion and runs those where the value reached zero and starts over again. Because the StartTimer and StopTimer are also actions, they can be put in any other region to run when the right trigger occurs. You can even put another StartTimer in the TimerRegion itself and

change the speed of the ticks on the fly (you don't need to stop the timer first). With this timer you are able to change a certain state after some time back to it's original, or run an animation every xx seconds, or switch to the next picture with some game content and so on... The timer can again stop with any other trigger from a region or also automatically after some ticks when located in the TimerRegion. The combinations are almost endless.

Every timerick creates an EVENT_GADGETPAINT event.

All actions of a SmartButton will use the same timer but can start at different times using a different interval.

SmartButton Creation:

Every SmartButton got now a 'SmartButton Creation' region which can hold actions which should run immediately after creation of the SmartButton itself. You can run an animation or a sound, or and that was the main reason for it, start the SmartButton timer without the need of a manually trigger for the first action. By this the SmartButton is able to do all actions automatically using the new creation region and timer intervals for following triggers.

RunActions:

A new function 'RunActions' in the SmartButton module allows you to start any action of a SmartButton and region. Thus you are able to start an action also from outside the SmartButton area, for example a button click can start an animation, a textfield change can trigger a new picture, or any other suitable event action combination...

Execute Function:

Own Function calls can be added as action to the SmartButton. Whith this possibility you are able to run your own code, whatever it is, on any trigger of a SmartButton region. Imagine a click in a certain region, maybe your logo, which calls your own function which in turn calls for instance a HTMLView gadget with a link to your website. A good place for this external functions are the append file in Logic Gui or perhaps a function in an imported file. You can enter the name of your own function below in the action properties panel. Enter only the name itself without (brackets).

Blur, Invert, Highlight, Color:

To have some more effects for the SmartButtons to visualize a certain state without the need of extra images you can use: First action blurs the current picture. Second action inverts the colors of the current picture. Third action highlights the current picture. Fourth action colors the current picture.

While the invert action has no options at all, the blur and highlight action have the ability to set a certain amount. The color action let you set separately the value for red, green and blue to get for instance a reddish picture. All values will be adjusted by sliders very easy.

All new actions are especially useful as a hover action, but can be used of course in combination with any other trigger.

Spinner:

There are a lot of different ways a spinner gadget could be useful, of course here with images for the SmartButton...

We have: SetSpinnerPicture, SpinnerForward and Spinnerbackward.

The spinner actions are separated for maximum flexibility. The SetSpinnerPicture action does set a series of pictures to the SmartButton, this can be done also at any time on any trigger on the fly, or if you don't want an extra trigger for that you can put the action of course in the creation region. The SpinnerForward and SpinnerBackward lets you then move to the next or previous picture. As those two actions are independent they can trigger on any available option. That means, SpinnerForward can trigger on wheel up and SpinnerBackward on wheel down, or you can use the left and right mouse button, or you can define two regions on a SmartButton and one cycles forward and the

other backward when clicked and so on...

The SpinnerForward and SpinnerBackward have one option to set, which is 'Infinite Loop'. When turned on and you reach one of the ends it starts again with pictures from the other end.