

Image Masks

What is an image mask?

Image masks are themselves images that are used to blend (or clip in the extreme case) a rendered (target) image.

How does image masking work?

During image rendering the mask may be thought of as overlaying the target image (the image you want rendered). Where the mask image has zeros in its alpha channel the target image will be rendered without any blending. If the image mask has a non-zero value in its alpha channel then the target image will be blended with the mask image. The blending occurs without regard for the background/canvas on which the target image is rendered.

What can I do with an image mask?

This capability was originally implemented to make it easy and more efficient to create images with rounded corners. However, image masking is very flexible due to the nature of alpha blending (and not just clipping) and so we expect that MediaPortal skin designers will be very creative in their implementations.

Is image rendering performance affected when I use image masking?

Image masking is implemented in the D3D graphics pipeline. If hardware acceleration is available then you will likely not notice any degradation in overall system performance when using this capability.

Is image rendering available for all controls that render images?

No, and yes. Extending controls that do not expose image masking is possible. Image masking is implemented in the core of the MediaPortal skin engine and is exposed primarily to the GUIImage control. Composite controls that use the GUIImage control can make use of the capability. This capability may only be exposed on several (not all) available skin controls. Adding the ability to mask images in other (composite) controls is relatively straightforward; it involves creating the interface on the skin control and passing the mask image value (the name of the desired mask image) into the GUIImage's that the control uses. In some cases the GUIImage is wrapped in an Animation or other class. In this case simply pass the mask image name through these class wrappers.