

1. FOP test for images

A GIF image in FOP

A transparent GIF image in FOP

A JPEG image in FOP

A BMP image in FOP

A PNG image in FOP

A TIFF image in FOP

This section is only required to show that the layout still works.

A. The usage of fo:external-graphic

The image used in these examples has the following properties:

169 x 51 px, 96 dpi, results in an intrinsic size of 44.715 x 13.494 mm

All the examples here will have a border="solid 0.1pt" to show the size of the viewport generated by the external-graphic element and a background-color="yellow" to show what parts of the viewport are painted upon by the image in the reference area. The image used is the non-transparent version of asf-logo.png: file:../graphics/asf-logo-nt.png (nt=non-transparent)

Properties: [none]

Properties: width="60mm" height="20mm"



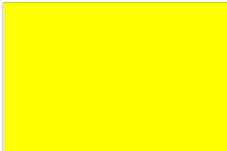
Properties: inline-progression-dimension="60mm" block-progression-dimension="20mm"



Properties: width="30mm" height="20mm"



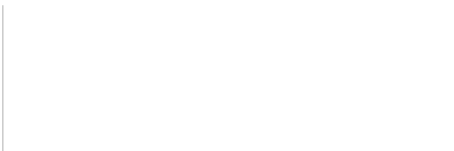
Properties: width="30mm" height="20mm" overflow="hidden"



Properties: width="60mm" height="20mm" text-align="center" display-align="center"

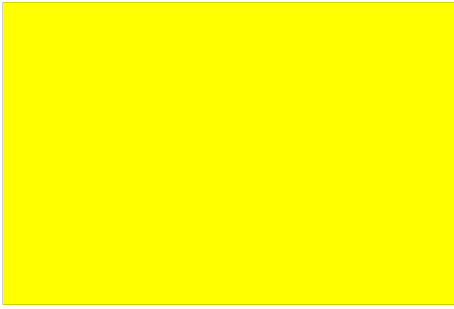


Properties: content-height="20mm"

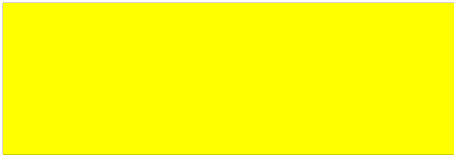


Properties: content-width="60mm" content-height="40mm"

Properties: content-width="60mm" content-height="40mm" scaling="non-uniform"



Properties: width="60mm" height="20mm" content-width="scale-to-fit"



Properties: width="60mm" content-width="scale-to-fit"

Properties: width="100%" content-width="scale-to-fit"

The following example shows how to fit an oversized image into the available width, but not scale smaller images up. This is a work-around until "scale-down-to-fit" is supported. The second feather below is in a block-container with a width of 40mm to simulate an oversized image.

Properties: width="100%" content-width="scale-to-fit" content-height="100%"

