

# Adobe® SVG Viewer for Windows

## Release Notes

**Version 3.0 (Build 76)**

## Contents

- [About Adobe SVG Viewer](#)
  - [System requirements](#)
  - [How to install Adobe SVG Viewer](#)
  - [How to install Netscape plug-ins for use with unsupported Browsers](#)
  - [How to view SVG files in your Web browser](#)
  - [How to turn artwork into SVG](#)
  - [Tips and Hints](#)
  - [Known Problems](#)
  - [New Features](#)
  - [Contact Information](#)
  - [Copyright Notice](#)
  - [More Information on Adobe's support for SVG](#)
- 

## About Adobe SVG Viewer

Adobe SVG Viewer 3.0 supports much of the W3C Recommended [Scalable Vector Graphics \(SVG\) 1.0 Specification](#) published on 4 September, 2001.

## System Requirements

- Windows 95, 98, SE, 2000, ME, XP, or NT 4.0 Service Pack 4 and up
- Netscape Navigator or Communicator versions 4.5 through 4.78 (but not 6.x), Internet Explorer 4.0 or higher, or RealPlayer 8 or higher.
- 13 MB of hard disk space
- 32 MB of RAM recommended

---

# How to install Adobe SVG Viewer

1. Before installing, please close any application which you have used to view SVG with an old version of Adobe SVG Viewer.
2. Run the installer. The latest released installers can be downloaded from <http://www.adobe.com/svg/viewer/install>.

# How to install Netscape plug-ins for use with unsupported Browsers

If you install another copy of Netscape which is supported by the Netscape plug-in, you can copy the following files into your plug-ins folder to view SVG from within that browser:

- NPSVG3.dll
- NPSVG3.zip

# How to view SVG files in your Web browser

Once you install Adobe SVG Viewer, you should be able to view supported SVG files in any supported Web browser.

# How to turn artwork into SVG

You can export SVG from Adobe Illustrator 9. If you wish to incorporate JavaScript into your SVG you can create your artwork in Illustrator and link to JavaScript functions with the JavaScript Interactivity Palette and then export it as SVG or export from Illustrator and add any JavaScript interactivity animation or filter effects by hand. You can place SVG into Adobe GoLive 5 software using the Preview Mode or use the GoLive Outline and Source Views to modify the SVG source code. You may for example add or modify JavaScript add or update links to images fonts or audio files or add or update SVG elements and their attributes.

# Tips and Hints

- Style attributes (or style attributes that use entity references) render more quickly than embedded stylesheets.
- Use tightly-bound percentages on the `x`, `y`, `width`, and `height` attributes on `filter` elements in order to minimize the area over which the filter effect needs to be calculated. This will result in graphics which display much more quickly.
- You can retrieve the SVG document object from within SVG event handlers by calling `getTarget` on the `evt` object which is passed in. For example:

```
// Retrieve the SVG document object:
var directTarget = evt.getTarget();
var svgDocument;
if( directTarget.getNodeType() != 9 ) // if not DOCUMENT_NODE
    svgDocument = directTarget.getOwnerDocument();
else
    svgDocument = directTarget;
```

- The Adobe SVG Viewer is capable of playing WAV and MP3 files linked via HTTP. This is the best way to play short sounds within the SVG presentation when embedded in RealPlayer®. However, long audio files should be streamed using the `RealAudio&REG;` format referenced from the SMIL document. For more information on Adobe's support for SMIL audio in Adobe SVG Viewer, please visit the Adobe [SVG Zone](#).
- Because the SVG animation model is open-ended, the timeline often can't be pre-determined. When playing in RealPlayer such animations will cause RealPlayer to display a very large (12+ hour) timeline to approximate the open-ended timeline. To avoid this problem, author the SVG such that its timeline ends at a desired time.
- The `bitrate` URL parameter is supported when playing SVG in RealPlayer. For example, if the URL

`foo.svg?bitrate=12000`

was used to play SVG in RealPlayer, the bitrate for the SVG packets would be 12000 bps (which happens to be the default bitrate).

- SVG is no different than any other datatype inside RealPlayer software, so there is no

# Known Problems with Adobe SVG Viewer

## All Browsers

1. It's possible for elements near the edge of an SVG graphic to not receive a mouseout event.
2. The SVG specification's support for masks has changed significantly since Illustrator 9 shipped, so Adobe SVG Viewer 2.0 and 3.0 might not display SVG masks exported from Illustrator 9.

## Internet Explorer

1. You must have Internet Explorer 4 or higher installed in order to use Adobe SVG Viewer as an ActiveX control in other applications (like Microsoft Office).
2. If you open more than one window containing a link to the same SVG file, and the SVG file is embedded using the HTML `<embed>` tag, then when you print one of the windows it may print based on changes to the DOM in the other window. This is due to limitations of Internet Explorer.
3. Due to Internet Explorer limitations, it's not possible to see properties of other windows from a script defined inside an SVG file.
4. Using the Adobe SVG Netscape plug-in under Internet Explorer on Windows is **not** supported.

## Netscape

1. Netscape 4.x plug-ins cannot execute `onunload` scripts because at the time the plug-in receives notification from Netscape that the plug-in is being unloaded, the plug-in's script environment has already been closed down.
2. Netscape 6.0 reports errors when printing pages with plug-ins, and then fails to print embedded contents. Netscape will report two plugin error alerts. Dismissing the alerts presented will then result in Netscape printing the HTML page, but with blank area for the SVG content.
3. When you leave a Web page, Netscape can often unload the Adobe SVG Viewer plug-in before Netscape terminates all scripts running in the HTML. Therefore you should be careful when writing scripts to check to see if the plug-in is still loaded before you attempt to access the SVG DOM from `setTimeout` or `setInterval` callbacks in your HTML JavaScript.

4. Due to design limitations of Netscape, avoid using HTML with the following form:

```
<a href="#" onclick="myScriptWhichModifiesSVG()">
```

The problem is that JavaScript responds to the onclick event and begins executing the script, but then Netscape also responds to the fact that you clicked on a link, and interrupts JavaScript to re-load the current page. This conflict can be avoided by re-writing the code as follows:

```
<a href="javascript:myScriptWhichModifiesSVG()">
```

5. If a JavaScript event handler contains an error, any subsequent JavaScript will fail to execute. This is due to a bug in Netscape's handling of scripts executed by plug-ins.
6. Due to Netscape script limitations, you can't change the URL of another frame from an SVG event handler.
7. Due to Netscape script engine limitations, calling the `alert()` function from within an SVG event handler might not work, and it might cause subsequent JavaScript to no longer execute. Similar problems occur for any Netscape modal window, including the security privileges window. This bug appears to have been partially fixed by Netscape in Netscape 4.75, but not on Windows 98.
8. There is a bug in Netscape involving plug-ins that use JavaScript. If you load a page with a plug-in that uses JavaScript and then disable Java and quit the browser, Netscape will crash.
9. Netscape does not always allow you to access the `document.embeds[]` array from within an SVG script. To work around this, use `document.mySVG`, where `mySVG` is the name of your embed object.
10. Errors in scripts executed by Netscape 4.x plug-ins do not show up in the Javascript console window.
11. Netscape 4.x leaks a small amount of memory every time a plug-in executes a script.
12. Netscape 4.x will crash when you close it if you launched it by double-clicking to load a Web page containing LiveConnect plug-in content.
13. If you use "Print Preview" in Netscape on a page which contains plug-ins (such as the Adobe SVG Viewer plug-in), Netscape will crash. For a similar reason Netscape will crash when you try to print a page containing plug-ins using Acrobat Distiller.
14. If you use Netscape 4.0x to view the same SVG file in two windows, and that file uses a global JavaScript function on the window object, sometimes JavaScript will not initialize in the second window. The work-around is to upgrade to a more recent version of Netscape.

15. Due to a bug in Netscape 6, attempting to right-click within the SVG area to bring up context menu fails in Netscape 6. Instead, you get the Netscape 6 context menu.
16. Netscape 4.x on Swedish systems has a bug where it does not identify its plug-in folder location correctly in the system registry, thereby making it impossible for installers to install plug-ins. You can work around this problem by copying the files by hand, as described [above](#).

## RealPlayer, standalone and embedded

1. Specifying the SMIL attributes `clip-begin` or `clip-end` in a SMIL element referencing SVG currently has no effect.
2. Specifying the URL parameters `start` or `end` within a URL to an SVG file currently have no effect.
3. RealPlayer software's autoupdate system won't match SVG content with the need to download the Adobe SVG Viewer Plugins for RealPlayer if the web server hosting the SVG content specifies the wrong SVG mimeType. Authors should be sure their web servers are using `image/svg+xml` as the SVG mimeType in order for RealPlayer software's autoupdate system to update the plugins appropriately.
4. When an author references SVG via RTSP, then URLs within the SVG must be absolute.
5. RealPlayer 8 on the Macintosh has a bug where it will crash when attempting to scroll plug-in content if the attribute `fit="scroll"` is specified on the `region` element in the SMIL file.

## RealPlayer, standalone

1. Using a favorites bookmark that contains an offset time for the start of a presentation involving SVG will generally fail due to a bug in RealPlayer software. It is not recommended that SMIL or SVG files be referenced by such bookmarks.
2. When viewing a SMIL document in fullscreen mode that references SVG alongside other datatypes, the SVG won't be centered correctly due to a bug in RealPlayer.

---

## New Features

### New features in Adobe SVG Viewer 3.0

- Elements and Attributes:
  - Added support for the following elements: `color-profile`, `marker`, `title`, and `view`.
  - Implemented full support for the `switch` element.
  - Implemented support for the general `requiredFeatures` and `systemLanguage` attributes.
- CSS properties:
  - Added support for the following CSS properties: `color-interpolation-filters`, `color-profile`, `marker`, `marker-end`, `marker-mid`, `marker-start`.
  - Added support for the `@media` CSS rule, and the `media` attribute for `style` elements. The values `all`, `screen`, and `print` are supported.
  - The `image` element now supports links to SVG files, but only static SVG files are supported in this way.
- DOM:
  - Added support for a large number of DOM interfaces, methods, and properties. Please see the current support documentation on the [Adobe SVG Zone](#) for details.
- Other Changes:
  - Performance improvements.
  - Adobe SVG Viewer now has its own built-in script engine. This allows self-contained JavaScript scripts to run inside of SVG files embedded in hosts which don't support a bridge between plug-ins and the host script engine, including Internet Explorer on the Mac, and RealPlayer. The internal script engine is automatically used in such environments, but it may be explicitly selected for use in all environments by specifying the attribute `scriptImplementation="Adobe"` on the top-level `svg` element. This attribute is in the Adobe SVG Viewer 3.0 extension namespace (<http://ns.adobe.com/AdobeSVGViewerExtensions/3.0/>). You may additionally over-ride this setting on a per-script basis by specifying the `scriptImplementation` attribute on a `script` element along with a `type` attribute. Supported `contentType` and `type` attribute values include `"text/ecmascript"`, `"text/javascript"`, and (on Windows only) `"text/jscript"`. Supported values for the `scriptImplementation` attribute include `"Adobe"`, `"Microsoft"` (Internet Explorer only), `"Netscape"` (Netscape only), `"browser"` (don't use Adobe's engine), and `""` (the empty string, and the default--try using the browser's script engine first, and if that fails use Adobe's). The Adobe SVG Viewer 3.0 internal script engine is based on source code governed by the [Mozilla Public License version 1.1](#). The modified versions of the source

files covered by the MPL [are available for the public](#) as required by the license.

- The Windows ActiveX control may now be invoked in a transparent mode which allows you to overlay SVG on top of other web page content. To invoke Adobe SVG Viewer in this mode, add the attribute `wmode="transparent"` to the `embed` tag. Other values for the `wmode` attribute include `windowed`, which is the default opaque (and faster) mode, and `opaque`, which is like `windowed` except that it supports z-order.
- Adobe SVG Viewer for Windows now implements support for Binary Behaviors in Internet Explorer 5.5 and higher.
- Animations will now continue to run even when the browser is not the foreground application.
- Added the ability to save as compressed SVG (.svgz).
- Added support for Macintosh OS 10.1, RealPlayer 8 and 9, and Windows XP.
- Support for ICC colors is no longer included as part of the default Adobe SVG Viewer download. To install the optional color management component, please visit the Adobe SVG Viewer [install page](#).
- If you set the `color-rendering` property to `optimizeQuality` on the top-most `svg` element in your SVG file, then all sRGB colors will be displayed in the device color space on screen if color management is installed and configured. However, this will slow down drawing and animation, so this functionality is disabled by default even if color management is enabled.
- Adobe SVG Viewer can improve the performance of some animations by generating sprites on demand during rendering. To invoke this functionality, you need to specify `shape-rendering="optimizeSpeed"` on any SVG graphics that you want to use sprites. Additionally, if there's text in the set of elements to be put in a sprite, you will need to also add either `text-rendering="optimizeSpeed"` or `text-rendering="optimizeLegibility"`. Since these settings have the additional effect of reducing quality (e.g. eliminating anti-aliasing), SVG authors will need to decide where it is acceptable to use these settings. There are a number of other conditions that will prevent a particular element from being drawn in a sprite:
  - The element has a non-transform animation (e.g. color changes)
  - The element uses a filter that references `BackgroundImage` or `BackgroundAlpha`
  - The element has opacity less than one
  - The element or an ancestor specifies the CSS property `adobe-`



knockout:true

- The Windows preferences are no longer stored in SVGViewer.ini, but instead they are now stored in the registry under  
HKEY\_CURRENT\_USER\Software\Adobe\Adobe SVG Viewer\3.0.
- 

## Contact Information

To report bugs and/or provide feedback, please go to the [SVG Zone](#) on Adobe.com.

## Copyright Notice

© Copyright 1998-2001 by Adobe Systems Incorporated. All rights reserved.

Adobe Illustrator and GoLive are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. SVG is a trademark (registered in numerous countries) of the World Wide Web Consortium; marks of the W3C are registered and held by its host institutions MIT INRIA and Keio. Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States and/or other countries. Mac and TrueType are trademarks of Apple Computer Inc. in the United States and/or other countries. RealNetworks, RealPlayer, and RealAudio are registered trademarks of RealNetworks. All other trademarks are property of their respective owners.