Developers Guidelines

June 2006

# SVG support in Sony Ericsson phones

Image created in Adobe® Creative Suite 2, courtesy of Adobe Systems Inc.

Sony Ericsson

# Preface

## Purpose of this document

This document is intended for developers who want insight into the implementation of SVG (Scalable Vector Graphics) in Sony Ericsson mobile phones.

People who can benefit from this document are:

- Content designers and developers
- Operators and service providers
- Content providers

It is assumed that the reader is familiar with the basics of Web graphics.

These Developers guidelines are published by:

Sony Ericsson Mobile Communications AB,
SE-221 88 Lund, Sweden

Phone: +46 46 19 40 00
Fax: +46 46 19 41 00
www.sonyericsson.com/

Eighth edition (June 2006)
Publication number: EN/LZT 108 7937, R8A

This document is published by Sony Ericsson Mobile Communications AB, without any warranty*. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Sony Ericsson Mobile Communications AB at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Printed versions are to be regarded as temporary reference copies only.

*All implied warranties, including without limitation the implied warranties of merchantability or fitness for a particular purpose, are excluded. In no event shall Sony Ericsson or its licensors be liable for incidental or consequential damages of any nature, including but not limited to lost profits or commercial loss, arising out of the use of the information in this document.

# Sony Ericsson Developer World

On www.sonyericsson.com/developer, developers will find documentation and tools such as phone White papers, Developers guidelines for different technologies, SDKs and relevant APIs. The website also contains discussion forums monitored by the Sony Ericsson Developer Support team, an extensive Knowledge base, Tips and tricks, example code and news.

Sony Ericsson also offers technical support services to professional developers. For more information about these professional services, visit the Sony Ericsson Developer World website.

# Document conventions

## Products

Sony Ericsson mobile phones are referred to in this document using generic names as follows:

| **Generic names**<br>Series | **Sony Ericsson mobile phones** |
|---|---|
| K700 | K700i, K700c |
| S700 | S700i, S700c, S710a |
| K500 | K500i, K500c, K506c, K508i, K508c |
| F500 | F500i |
| Z500 | Z500a |
| V800 | V800, Vodafone 802SE |
| Z800 | Z800i |
| K300 | K300i, K300c, K300a |
| K750 | K750i, K750c, D750i |
| W800 | W800i, W800c |
| K600 | K600i, K608i |
| V600 | V600i |
| Z520 | Z520i, Z520c, Z520a |
| W600 | W600i |
| W550 | W550i, W550c |
| W900 | W900i |

| **Generic names**<br>Series | **Sony Ericsson mobile phones** |
|---|---|
| W810 | W810i, W810c |
| K610 | K610i, K610c, K610im |
| K800 | K800i, K800c, K800a |
| K790 | K790i, K790c, K790a |
| Z530 | Z530i, Z530c |
| W300 | W300i, W300c |
| K510 | K510i, K510c |
| W700 | W700i, W700c |
| Z525 | Z525a |
| Z550 | Z550i, Z550c |
| W850 | W850i, W850c |
| Z710 | Z710i, Z710c |
| W710 | W710i, W710c |

# Trademarks and acknowledgements

Microsoft and Microsoft Windows are either registered trademarks or trademarks of Microsoft Corporation in the Unites States and/or other countries.

Mac OS is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.

Adobe, Adobe Golive, Adobe Illustrator and Adobe Photoshop are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

# Document history

| Change history | | |
|---|---|---|
| 2005-04-11 | Version R1A | First version published on Sony Ericsson Developer World. |
| 2005-05-04 | Version R1B | First revised version. |
| 2005-11-07 | Version R2A | Second version. |
| 2006-01-05 | Version R3A | Third version. |
| 2006-02-13 | Version R4A | Fourth version. |
| 2006-02-28 | Version R5A | Fifth version. |
| 2006-04-04 | Version R6A | Sixth version. |
| 2006-05-19 | Version R7A | Seventh version. |
| 2006-06-19 | Version R8A | Eighth version. |

# Contents

# Scalable Vector Graphics

SVG (Scalable Vector Graphics) is an open standard format for vector graphics on the Web.

# SVG overview

## Introduction

In 2001, the World Wide Web Consortium (W3C) approved Scalable Vector Graphics (SVG) as its standard for XML (eXtensible Mark-up Language) vector graphics on the web.  Because it's based on XML and is an open standard, SVG has become a component of many industry initiatives and Enterprise solutions. As large parts of SVG are modeled after HTML, web developers find SVG to be the perfect complement to HTML. HTML and SVG are used side-by-side in web applications, with HTML as the language for text-oriented applications and SVG as the language for graphics-oriented applications.

SVG is the open-standard, XML-based graphics that leverages other standards such as CSS (Cascading Style Sheets), Synchronized Multimedia Integration Language (SMIL), ECMA-script, javascript and the Document Object Model (DOM) to provide a powerful common platform able to span everything from desktop web applications to embedded devices as well as mobile phones from Sony Ericsson. As a text format, open-standard SVG is easily scripted and can be authored and dynamically generated using a number of popular design and development tools. The SVG specification itself is regulated by the World Wide Web Consortium (W3C) and includes members from leading companies such as Adobe®, Apple, Canon, Ericsson, HP, Nokia, Sun and Ikivo. SVG comes in three profiles, Full (for desktop applications), Basic (for portable devices like PDAs), and Tiny (for mobile phones).

The mobile subset of the SVG specification, called SVGT or SVG Tiny, has been adopted by the 3GPP, OMA (Open Mobile Alliance) and other leading mobile-standardization bodies. The most notable difference between SVG Tiny and the full SVG profile is the limited interactivity support.

The primary benefit of vector-based graphics over bitmap images such as GIF and JPEG is that vector-based images can be scaled without the loss of image quality.  SVG content can be used for animated cartoons and interactive graphics.

Another benefit of SVG is that vector-based images are more compact than bitmap images. This is because they are built up by geometrical shapes that are described mathematically, rather than by bitmaps. SVG images, which are based on XML, are in addition compressed with standard gzip-compression. This makes vector-based images attractive for mobile phones and wireless networks.

Vector-based graphics are inappropriate to use for pictures and other kinds of images that are not composed of distinct lines and curves. A bitmap image, such as a GIF, can however be embedded inside the vector-based image. In practice, animations are often created by using a bitmap in the background and animated vectors in the foreground.

Since SVG is based on XML, it is particularly good for data driven graphics (where the data is delivered in XML from the backend), such as weather services, stock quotes, and statistical data. Another advantage is that an SVG image is searchable, which opens up new possibilities for search engines (for example Google), and makes SVG particularly appropriate for maps.

# SVG support in Sony Ericsson handsets

All handsets in the scope of this document support viewing of SVGT content received via MMS.

The web browser in the W600, W550, W900, W810, K610, K800, K790, Z530, W300, K510, Z550, W850, Z710 and W710 series also support SVGT in web content and viewing of downloaded and locally stored SVG files.

The K750, W800, Z520, K600, V600, W600, W550, W900, W810, K610, K800, K790, Z530, W300, K510, W700, Z525, Z550, W850, Z710 and W710 series ,allow use of SVGT content as wallpaper images selected in the display settings menu.

The K750, W800, Z520, K600, V600, W810, K610, K800, K790, Z530, W300, K510, W700, Z525, Z550, W850, Z710 and W710 also allows use of SVGT 1.1 content in the Standby (wallpaper) element of Sony Ericsson themes.

The evolution of SVG features support for different Sony Ericsson mobile phone series is described below.

| | **K700, K500, F500, Z500 S700, K300** | **V800** | **Z800** | **K750, W800, Z520, K600, V600, W600, W550, W900, W810, K610, K800, K790, Z530, W300, K510, W700, Z525, Z550, W850, Z710, W710** |
|---|---|---|---|---|
| **SVGT 1.1** | • | • | • | • |
| **Links** | | • | • | • |
| **Opacity** and **gradients** (addition to SVGT 1.1) | | | • | • |
| **Anti-aliasing** | | | • | • |
| **System fonts** (bitmaps) | | | | • |

**SVG Tiny version 1.1** (SVGT 1.1) is the standard SVG version for mobile phones.

**Links** lets the user interact with the image to activate hyperlinks to other Web resources. It is possible to use the SVG image as a Web page.

**Opacity** and **Gradients** are additional SVG features, not supported in the SVGT 1.1 specificaton, but in the 1.2 version. Sometimes SVGT 1.1 with extended support for opacity and gradients is referred to as SVGT 1.1+.

**Anti-aliasing** is a feature in the SVG player in the phone. It removes jagged lines and makes objects look smoother.

**System fonts** mean that the SVG author is able to use the system fonts of the phone to write text in images. Without this feature, fonts must be provided as part of the SVG image, so called SVG fonts.

# Creating static SVGT content

## General tips for creating static SVGT content

Generally, when creating SVGT 1.1 content, text objects should be converted to paths. Font support on phones is varied, and some phones do not support SVG text at all, see table above. To ensure that text is visible and displayed in the font-face that you desire, on the widest range of SVG-enabled handsets, you should convert your text objects to paths.

Note that although SVG emphasizes vector content, raster images, such as GIFs, JPEGs, and PNGs are supported. In some cases, using a raster object may be a more optimal solution than using a vector object. The only drawback is that the image will be less scalable. (For more about using raster content, see the section below on General SVG optimization tips).

To keep resorces as lokalized as possible, rasterized content should always be embedded instead of referenced by file name, thus avoiding problems for the rendering software to find referenced files . Raster images to embed should be optimized by resizing the image and using an optimization program (such as the Adobe® Photoshop® "Save for Web" function), and then place the image into the SVG graphic tool used. The "embed images" option should be selected when saving the SVG.

There are several ways to convert SVG Full content to SVG Tiny. For example the following methods are useful:

- Generate SVG Tiny content with SVGmaker Tiny.
- Open the illustration in Adobe® Illustrator® CS2 and save it using the SVG Tiny 1.1 or 1.1+ DTD.
- Open or insert the content into Ikivo Animator to convert or remove non-Tiny compliant content and save it as SVG Tiny 1.1+ (see more about the Ikivo Animator below).
- Open the file in Adobe® GoLive® (CS1 or CS2) and Check Syntax against SVG Tiny 1.1. Make the suggested changes based on error messages until the file is compliant.

## Graphic tools

Several commercial vector-based graphic tools are well suited to generate SVG files. Most of these are not specific to SVG Tiny creation, but with a few tweaks, the output can be constrained to SVG Tiny limitations.

### Adobe Illustrator CS2 (Mac OS® and Microsoft® Windows®)

Adobe Illustrator versions 10 and up can export SVG, but the support for SVG has been greatly enhanced in the Creative Suite (CS) versions.

For detailed information about Adobe Illustrator, please refer to http://www.adobe.com/products/illustrator/main.html

The following Adobe Illustrator features are useful for SVGT content creation:

- SVG 1.0, 1.1, and 1.1+ Full, Basic, and Tiny profile export.
- Conversion of text to outlines (ensures proper display of text in devices with limited font support).
- LiveTrace functionality allows creation of vectors out of photographic or other raster images.
- Bitmap embedding (embedded bitmap data ensures proper asset management - no missing links)
- Support for XML IDs
- Round-trippable SVG (import and preserve SVG animation elements)

### Adobe Illustrator tips

- For the most streamlined SVG export, make sure that the following items in the SVG Options dialog are **not** checked (click *More Options* to see all items):
  - Preserve Illustrator Editing Capabilities
  - Optimize for Adobe SVG Viewer
  - Include Adobe Graphics Server data
  - Include Slicing Data
  - Include XMP

- To minimize the path data points when using Live Trace:
  - Minimize number of colors used (fewer colors gives fewer paths)
  - Maximize area for sampling (the larger "Min. Area" value, the less the path data)

## CorelDRAW Graphics Suite 12 (Microsoft Windows only)

For detailed information about CorelDRAW, please refer to http://www.corel.com

The following CorelDRAW features are useful for SVGT content creation:

- Round-trippable SVG (import and preserve SVG elements)
- Conversion of text to outlines (ensures proper display of text in devices with limited font support)
- Preservation of object names as IDs
- Bitmap embedding (embedded bitmap data ensures proper asset management - no missing links)
- CorelTRACE allows creation of vectors out of photographic or other raster images

### CorelDRAW tips

- CorelDRAW does not have an explicit SVG Tiny export option. To conform as closely as possible to the SVG 1.1+ Tiny DTD:
  - Set "Styling Options" to "Presentation Attributes"
  - Export Text "As Curves"

- The size of the illustration is as indicated by the page size under "Page Setup".

- When working with a multi-page layout, only the current page is exported as SVG.

- When renaming an object, XML naming conventions must be followed:
  - Names cannot contain spaces, and cannot begin with a number
  - Names must begin with a letter, an underscore (_) or colon (:)

### CorelTRACE tips

- When using "Trace by Advanced Outline", path complexity should be minimized by setting:

- highest possible value for "Noise Filter".
- lowest possible value for "Complexity".
- minimum number of colors.
- lowest number of nodes.

## SVGmaker Tiny (Microsoft Windows only)

SVGmaker Tiny is a printer driver for generation of SVGT content from any Microsoft® Windows® program. One of its most important features is that it converts text to paths.

For detailed information about SVGmaker Tiny, please refer to http://www.svgmaker.com/

### SVGmaker Tiny tips

- SVGmaker Tiny should be used, not SVGmaker Standard

- Using SVGmaker Tiny will not group path objects that are generated when text objects are converted to paths. However, you may find that many letters are joined as compound paths (particularly the lower-case string in a word), and they are "printed" to the SVG document from the graphical layout in order from top to bottom, left to right. This should be considered when editing SVGmaker output in other programs.

- IDs are lost or replaced when content is run through SVGmaker Tiny.

## Inkscape (Mac OS, Microsoft Windows, Linux)

Inkscape is an open source SVG drawing tool.

For detailed information about Inkscape, please refer to http://inkscape.org/

Inkscape features:

- Lots of online tutorials and documentation contributed by users.
- XML editor for source code editing.
- Trace Bitmap command allows creation of vectors out of photographic or other raster images.

# Creating SVG animations

SVG animations follow simple instructions over time. This is in contrast to frame-based animation formats like Macromedia Flash (SWF), where each frame of an animation must be drawn. Time-based animations like SVG animations are highly efficient, and are not dependent on processor speed. For example, a 5 second animation will last exactly 5 seconds, and an animation requires no more code to run for 5 hours than for 5 seconds. Time-based animation ensures precise synchronization with other media and events such as the start or end of sound and video.

It is important to note that SVG animations are based on the SMIL (Synchronized Multimedia Integration Language) Specification. As a result of that, timing of animations takes precedence over display, animations must render accurately according to the timeline. When rendering frame-based (cartoon-type) animations in a SVG viewer, the viewer may have to drop some frames to force the animation to fit into the time-slot assigned to it. Therefore, it is strongly recommended to limit the use of frame-based animations as far as possible, and simply animate color fills or position/rotation/scale transformations instead.

## Ikivo Animator 1.1 (Mac OS and Microsoft Windows)

For detailed information about Ikivo Animator 1.1, please refer to
http://www.ikivo.com/animator/index.html

Ikivo Animator 1.1 features:

- Imports and opens any SVG formats (Tiny through Full profiles) and outputs SVGT 1.1+ content.
- Includes Ikivo SVG Player software for accurate preview of animations using real device profiles (emulating color depth, frame rate and anti-aliasing). Device profiles are available for all Sony Ericsson phones with SVG support.
- Preserves SVG animations created in other tools, but not understood by Animator (because the SVG Specification allows different ways of describing the same animation, and Ikivo Animator can only comprehend a limited range of these variations, Trapping enables the preservation of other, valid forms of SVG animation)
- Integration with Adobe Illustrator CS2 and Adobe GoLive CS2
- Includes professional, but easy to use, creative palettes for color, transformation, and paced (ease-in/ease-out) animations

Resources in the form of tips and tricks, developer forums, and tutorials for the Ikivo Animator and Ikivo SVG Player can be found here:

- Ikivo Animator Mobile Developers Zone. Online tutorials covering a variety of animation features in Animator
  http://www.ikivo.com/animator/devzone/index.html

- Ikivo Animator Workflow Tutorial. Downloadable PDFs for step-by-step instructions on how to build a sample location based services-type map application (Mac OS and Windows versions).
  http://www.ikivo.com/animator/02animator_7downloads.html

# General SVG optimization tips

- If there are multiple copies of an object in an animation, the same object should be used wherever possible.

  For example, instead of having 10 different cars passing by, the minimum number appearing in the scene at any time should be used, and their position, visibility, and so on, can be animated and still seem to appear like 10 different objects.

- SVG animations, as well as other properties and attributes, are inherited from parents (groups) to their children (objects in the group). This means that an animation applied to a group will also apply to the children. This should be kept in mind when animating multiple objects with the same animation. Depending on the structure of the illustration, objects may be combined into a single group, and only the parent may need to be animated instead of each child.

- Combine path objects to make compound paths whenever possible

  For example, if text elements have been converted into paths, separate letters should be combined into a single compound path. As opposed to a group of paths, a compound path is a single definition of a path with multiple end points (refer to the user manual of the creative application in use). By reducing multiple path objects into a single path object, both the file size and the number of animations needed may be reduced.

- Some animations require more processor power than others, alternative methods should always be tried.

  Instead of animating the opacity of an object, it may be better to animate the fill and/or stroke colors to a background color (for example, animate a red rectangle on a blue background to blue instead of to 0 opacity)

- Sometimes rasters are better than vectors.

  If it is certain that a graphic object will not need to scale, or that the image quality will not be compromized in the animation, it might be desired to use a GIF, JPEG, or PNG instead. It depends on the complexity of the vector description as opposed to the complexity of the pixel-per-pixel definition of an image. For example, a photographic image with many colors, gradients, and path points, may be simpler to describe as a raster JPEG image. However, objects which are originally vectors, such as logos or geometric objects, may be more efficiently described as vectors. It is advised to test different approaches and see which works best for the content, both in terms of performance and file size.

  Rasterized images should also be used for content with complex graphic effects, such as blurs and mesh-type gradients. Cool animation effects can be created on objects like these.

- Reduce path points to the minimum required.

  Removing unnecessary path points can greatly reduce the amount of data without compromizing the image quality (see notes above on using Adobe Illustrator's LiveTrace and Corel's CorelTRACE tools).

- Gradients are processor intensive and should be used with some care. Especially gradients used over large areas could effect the performance in a negative way.

- Use GZIP compression to produce compressed SVG files.

  GZIP compression, available with various freeware and shareware utilities, is the standard compression for SVG (called SVGZ). Depending on the size of the file, and whether there are embedded raster images in the file, up to 90% compression can be achieved.

# Links and references

For more information about SVG:

- www.w3.org/Graphics/SVG - the official SVG specification web site
- www.svg.org - a community web site about SVG
- www.adobe.com/svg/ - Adobe SVG zone
- www.ikivo.com/animator/ - mobile SVG content creation tool
- www.beatware.com/products/md.html - mobile SVG content creation tool