SVG in the Mobile Domain Requirements

Candidate Version 1.0 – 12 Jan 2005

Open Mobile Alliance

OMA-RD-SvgMobileDomain-V1_0-20050112-C

Contents

1. Scope (Informative)

This document contains the requirements for adopting Scalable Vector Graphics (SVG) in the domain of mobile devices.

Detailed use cases are presented along with the resulting functional requirements for the mobile device supporting SVG. In addition, key conformance and interoperability considerations are presented, aimed at those implementing these SVG requirements on a mobile device.

The goal of this document is to aid the mobile developer community in creating SVG content which is interoperable across different devices in the mobile space.

2. References

2.1 Normative References

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[SVGT1.2] “Mobile SVG Profiles: SVG Tiny and SVG Basic” Working Draft, W3C, December 2003, http://www.w3.org/TR/SVGMobile12

[JSR226] “JSR 226: Scalable 2D Vector Graphics API for J2ME”, http://jcp.org/en/jsr/detail?id=226

[CRS] “SVG Geographic Coordinate Systems”, http://www.w3.org/TR/SVG11/coords.html#GeographicCoordinates

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Abbreviations

OMA	Open Mobile Alliance
W3C	World Wide Web Consortium
SVG	Scalable Vector Graphics
SVGT	Scalable Vector Graphics Tiny (mobile profile)
GPS	Geographic Positioning System
CRS	Coordinate Reference System, an OpenGIS recommendation. See [CRS]

4. Introduction (Informative)

Scalable Vector Graphics (SVG) as defined by W3C is a language for describing two-dimensional graphics and graphical applications in XML. There are numerous use cases such as pan-able and zoom-able maps, 2D animations, remote rendered documents and location information services that make SVG particularly applicable to the mobile environment. "It has been determined by the W3C that some form of SVG suited to mobile devices is required. Consequently, W3C is now in the process of specifying Mobile SVG Profiles to render SVG on mobile devices with limited memory, CPU power, and bandwidth - SVG Tiny, targeted for highly restricted mobile devices; and SVG Basic targeted for more capable mobile devices. These profiles introduce constraints on content, attribute types, properties and user agent behavior and consist of certain modules of the W3C SVG 1.1 Recommendation of 14 January 2003 - SVG Tiny is specified as a proper subset of SVG Basic, and SVG Basic as a proper subset of SVG 1.1, so that SVG 1.1 can be transcoded into SVG Tiny and SVG Basic preserving as much scalability as possible. "

W3C is now in the process of defining Mobile SVG Profiles: SVG Tiny and SVG Basic, version 1.2, as per the second public working draft of 25 March 2004. " In order to guarantee that these profiles meet the requirements of OMA members, it is important to define OMA requirements and liase with W3C to insure OMA requirements are met.

OMA requirements for SVG fall into four main areas:

1. What is the subset of features of full SVG 1.2 that are appropriate for the mobile environment and resource limited devices.

2. What are the interaction modes needed to support SVG within the XHTML-MP document environment?

3. What are the requirements on ECMAScript-Mobile Profile (ESMP) and DOM, when managing and manipulating SVG documents?

4. What are the requirements on Wireless Cascading Style Sheets (WCSS) when supporting an SVG profile?

OMA user agents that may support SVG are the browser user agent and the MMS user agent. Furthermore, it is known that the Java Community Process is in the process of specifying "Scalable 2D Vector Graphics API for J2ME".

Lastly there are conformance considerations for SVG on mobile devices that are outlined in Appendix B. These conformance issues should be taken into consideration by implementers to ensure interoperability.

5. Use Cases (Informative)

The following sections discuss the various use cases for SVG in the mobile domain.

5.1 Use Case, Map Visualization and Interaction

	Affected Areas				
	Device	Connectivity	Enabling Services	Applications	Content
Tickmarks (X)	X		X	X	X
Additional Keywords				Map Viewer	text/svg

Table 1: Affected Areas for Map Visualization and Interaction

5.1.1 Short Description

SVG has a tremendous potential for enhancing the current user experience around map visualizations and applications. Some sample applications of appropriate mapping applications include things like navigation (e.g. driving directions), informational (e.g. restaurant guides, tourist information), or safety (e.g. traffic alerts, weather). The use of SVG for mapping applications does not have to restrict itself to maps of land regions. For example a user could be presented with a layout of a stadium in SVG and select the seats to purchase for a concert.

The benefits of using SVG for mapping visualization and applications are many. First, since the content is vector based, one map can be zoomed/panned/rotated by the user without any loss in image quality or without having to fetch any new information from the server. Second, with the web-based layering function (Hyper-Layering), portions of the map can updated without having to fetch the whole map. For example new traffic alerts can be added to an existing map as they come in. Third, since SVG is interactive the maps can be interactive. Fourth, since SVG is XML based the maps can be easily searched for text or image properties. Fifth, SVG has a function to associate itself with a Geographic Coordinate System and therefore data of positioning systems such as GPS can be plotted onto maps.

5.1.2 Actors

End User – individual using the mapping application

Service Providers – the organizations that are offering mapping applications to the End User

Content Providers – the organizations that provide geographic and map related data to the Service Provider

5.1.2.1 Actor Specific Issues

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must be willing to convert map data to SVG if the content provider is not willing to do it for them.

Content Provider – Must be willing to provide their map data to be converted to SVG. This is the ideal case. If the content provider is not willing to provide SVG content then they must at least allow their content to be easily converted to SVG.

5.1.2.2 Actor Specific Benefits

End User – The End User is provided with a richer and more fulfilling experience. Since the SVG map can be zoomed, panned, etc without having to re-fetch the map from the server the user is provided with a more responsive user interface. Since SVG content can be freely rotated the map can be rotated on screen to match the user’s orientation. Also the maps can be interactive and animated to ease tasks like following navigation directions. In addition, the End User can obtain locations of himself and his friends by plotting information measured by positioning systems like GPS onto maps. Contents of several service providers can be viewed at once.

Service Provider – More people would be willing to pay for high quality maps like the SVG ones described in this use case. Also more people would use this service with these benefits.

Content Provider – Since maps do not have to be re-fetched as often the Content Provider will not have such an intense load on their application servers. (when compared to mapping applications that use raster images to display map data)

5.1.3 Pre-conditions

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must be willing to convert map data to SVG if the content provider is not willing to do it for them.

Content Provider – Must be willing to provide their map data to be converted to SVG. This is the ideal case. If the content provider is not willing to provide SVG content then they must at least allow their content to be easily converted to SVG.

5.1.4 Post-conditions

The End User has use of the maps provided by the Content Provider.

5.1.5 Normal Flow

The following flow shows the interaction for a simple request for a restaurant and directions:

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
1. The End User wants a particular type of restaurant.							
	2. The Service Provider receives the request and looks up the End User's current location. The Service Provider searches for a combination of contents providers that can form a map with restaurants information.						

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
	The Service Provider returns the SVG Map Contents referencing contents of information on restaurants and contents provided by Base Map Providers.						
The Browser requests for referenced contents to the base map providers and the Contents Providers of information on restaurants.							
		3. The Service Provider asks the Content Provider to give information on all appropriate restaurants within the End User's area.		3. The Service Provider asks the Content Provider to give information on all appropriate basemap within the End User's area.			

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
			4. The Content Provider creates the appropriate SVG map with the restaurants shown and sends it back.		4. The Content Provider creates the appropriate SVG map with the restaurants shown and sends it back.		
		5. The Service Provider receives the SVG map from the Content Provider and does any necessary post processing before sending it back to the End User.		5. The Service Provider receives the SVG map from the Content Provider and does any necessary post processing before sending it back to the End User.			
6. The End User receives the SVG maps from the Service Providers and views it.							
7. The End User interacts with the map (zooming, panning, etc.) and finds the restaurant they want to go to.							

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
In addition, the End User wants to know the location of ATM and queries it.	Do a search for contents provider of ATM in addition and return SVG contents that reference the basemap providers and contents providers of restaurant and ATM information contents.						
The browser issues a new request only to an ATM contents provider as it reuses downloaded contents. (Omit ATM contents providers)							
Implement layering of the obtained ATM contents.							
The End User can now read a map with information on restaurants and ATM.							

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
The browser plots the location of user on a map based on the data generated by the positioning system. (Alternatively, scroll the map automatically to set the location of the user in the center. If a compass in hand, head-up display is possible.)							
8. The End User requests navigation directions to the restaurant. The Browser issues a new request only to the Navigation Service Provider, as it reuses the contents that are already downloaded.	The Service Provider returns SVG contents that further include references to the Navigation Server.						

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
						9. The Service Provider receives the request and looks up the End User's current location.	
						10. The Service Provider asks the Content Provider to create SVG directions from point A to point B.	
							11. The Content Provider creates the appropriate SVG directions map and sends it back.
						12. The Service Provider receives the SVG directions from the Content Provider and does any necessary post processing before sending it back to the End User.	

End User	**Service Provider** (Yellow Pages)	**Service Provider** (thematic contents (restaurants))	**Content Provider** (thematic contents (restaurants))	**Service Provider** (Base Map)	**Content Provider** (Base Map)	**Service Provider** (Navigation)	**Content Provider** (Navigation)
13. The end user receives the SVG directions from the Service Provider and implement layering. Then the End User uses it to go to desired location.							

5.1.6 Operational and Quality of Experience Requirements

MAP 01 Techniques for navigating the map should be convenient and natural

MAP 02 Maps must be able to be zoomed/panned and maintain their visual quality

MAP 03 Maps should be able to be rotated and maintain their visual quality

MAP 04 Maps should not be too large; this would prohibit transmissions via MMS

MAP 05 Maps should be legible at various resolutions.

MAP 06 Complex path data must be supported to render map components (i.e. roads, etc)

MAP 07 Graphics content should be able to be rendered over raster images

MAP 08 Scalable fonts must be supported to render map details properly

MAP 09 Must be able to render text at various angles

MAP 10 Must support a way to reuse user-defined content in a single file (i.e. define a visual element and instruct it to be rendered multiple times, for example a freeway sign on a map)

MAP 11 CRS should be supported

MAP 12 Hyper-Layering function should be supported

MAP 13 Scale based display control should be supported

5.2 Use Case, Animations (MMS, cartoon delivery, etc)

	Affected Areas				
	Device	Connectivity	Enabling Services	Applications	Content
Tickmarks (X)	X		X	X	X
Additional Keywords				SVG Viewer	text/svg

Table 2: Affected Areas for Animations (MMS, cartoon delivery, etc)

5.2.1 Short Description

SVG has a set of rich capabilities for describing animations and interaction. This is ideal for providing the user with a polished experience with MMS or other delivery mechanisms. Sample use cases include cartoon delivery, event announcements, etc. The content could either be user created or purchased.

The benefits of using SVG for this use case center around the efficient set of animation and interaction features of SVG. Lengthy (time wise) animations can be packed into small deliverables.

5.2.2 Actors

End User – individual enjoying the animations

Service Provider – the organization that is offering applications to the End User

Content Provider(s) – the organization(s) that create the graphic content or own the rights to the characters in the animation

5.2.2.1 Actor Specific Issues

End User – Must have a mobile device that can display SVG content.

Content Provider – Must be willing to provide their animations in SVG.

5.2.2.2 Actor Specific Benefits

End User – The End User is provided with a richer and more fulfilling experience. SVG based animations are more efficient than other forms of cartoon delivery. The content can be interactive and support things like “quiz games” or “what would you do”.

Service Provider – People would be willing to pay for an animated daily cartoon delivery service like this.

Content Provider – The more people that would view their cartoons the more brand recognition they would obtain.

5.2.3 Pre-conditions

End User – Must have a mobile device that can display SVG content.

Content Provider – Must be willing to provide their animations in SVG.

5.2.4 Post-conditions

The End User has the animations on their mobile device and gets to enjoy them.

5.2.5 Normal Flow

The following flow shows the interaction for a simple cartoon a day type of service:

End User	Service Provider	Content Provider
1. The End User registers for a cartoon a day type of service.		
	2. The Service Provider receives the request and adds the End User to the list of subscribers.	
… time passes …		
		3. The Content Provider makes the daily animation available to the Service Provider.
	4. The Service Provider receives the new animation from the Content Provider.	
	5. The Service Provider sends out the new cartoon to each subscriber over the desired means.	
6. The End User receives the latest animation via MMS (since this is their desired way to receive these animations).		
7. The End User enjoys the animation on their mobile device.		

The following flow shows the interaction for the user receiving an interactive animation.

End User	Service Provider	Content Provider
		1. The Content Provider creates a new interactive animation promoting a new movie (or other product).
	2. The Service Provider distributes the new interactive animation to all users who are interested in hearing new movie announcements.	
3. The End User receives the new interactive animation via MMS.		
4. The End User watches and interacts with the content on their mobile device.		
5. The user goes to see the new movie the following weekend.		

5.2.6 Operational and Quality of Experience Requirements

ANI 01 Content should be efficient (SVG is more efficient than animated GIF)

ANI 02 Interacting with animations should be done in a natural way and be usable on mobile devices

ANI 03 Complex path data must be supported to render interesting content

ANI 04 Graphics content should be able to be rendered over raster images

ANI 05 Scalable fonts must be supported to render things like dialog bubbles effectively

5.3 Use Case, Scalable User Interface Components (icons, controls, etc)

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)	X		X	X	X
Additional Keywords				SVG Engine	text/svg

Table 3: Affected Areas for Scalable User Interface Components (icons, controls, etc)

5.3.1 Short Description

One of the problems facing UI and content designers today is the multitude of devices on the market today. There are many devices with varying display capabilities. SVG could be used to provide a totally scalable, skin-able, and flexible UI framework to be used across devices. For example icons and UI controls could be done in SVG and their size could vary depending on the device capabilities.

5.3.2 Actors

End User – individual with a SVG enabled mobile device

Content Provider(s) – the organization(s) that create the graphic content for each UI and skin

5.3.2.1 Actor Specific Issues

End User – Must have a mobile device that uses a SVG enabled UI

Content Provider(s) – Must be willing to provide their content as SVG

5.3.2.2 Actor Specific Benefits

End User – The End User is provided with a richer and more fulfilling experience. A SVG based UI can be more customisable and visually more appealing. Other End Users could create UI skins and upload them to their mobile device if desired.

Content Provider – People would be willing to buy new skins to change the look and feel of their phone. The Content Provider could design a skin once and use it on multiple devices that have varying display resolutions and properties. Since the same content can be reused there would not be a lot of work to support a particular skin on a large number of devices.

5.3.3 Pre-conditions

End User – Must have a mobile device that uses a SVG enabled UI

Content Provider(s) – Must be willing to provide their content as SVG

5.3.4 Post-conditions

The End User has the new UI on their mobile device and gets to enjoy the new look of their device's interface.

5.3.5 Normal Flow

The following flow shows the interaction for an End User downloading a new UI skin:

End User	Content Provider
	1. The Content Provider creates a collection of different skins for varying mobile devices.
2. The End User comes across a particular skin from the Content Provider.	
3. The End User requests the content be sent to their phone for install.	
4. The End User installs the new skin and enjoys it.	

5.3.6 Operational and Quality of Experience Requirements

SUI 01 Content should be efficient

SUI 02 New skins should be usable and not break or hide any functionality of the device

SUI 03 Complex path data must be supported to render interesting content

SUI 04 Graphics content should be able to be rendered over raster images

SUI 05 Scalable fonts must be supported for flexible text layout

5.4 Use Case, Document Sharing and Portability

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)	X		X	X	X
Additional Keywords				SVG Engine	text/svg

Table 4: Affected Areas for Document Sharing and Portability

5.4.1 Short Description

SVG can be used instead of proprietary formats for document sharing and distribution. For example there are a large number of phones that are capable of displaying SVG (either out of the box or with extra software). If a standard format like SVG is used then documents like MS Word or Power Point can be easily distributed and shared across multiple devices and platforms.

5.4.2 Actors

End User – the individual(s) that desire to share the documents

Service Provider – the organization that is offering the transcoding service to the End User(s)

5.4.2.1 Actor Specific Issues

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must be willing and capable to convert desired documents to SVG for the End User.

5.4.2.2 Actor Specific Benefits

End User – The End User is provided with a richer and more fulfilling experience. Documents can be exchanged in a standard format (SVG) and shared across devices with varying display properties.

Service Provider – More people would be willing to pay for a service like this.

5.4.3 Pre-conditions

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must be willing and capable to convert desired documents to SVG for the End User.

5.4.4 Post-conditions

The End User(s) can share and view the documents on their mobile device or desktop computer.

5.4.5 Normal Flow

The following flow shows the interaction for a simple document sharing session:

End User	Service Provider
1. User A sends a PowerPoint document to User B via MMS.	
	2. The transcoding middleware converts the PowerPoint file to SVG and forwards to User B.
3. User B receives the SVG file (previously PowerPoint file) and views it on their mobile.	

5.4.6 Operational and Quality of Experience Requirements

DOC 01 Content should be efficient

DOC 02 Documents must maintain their original appearance

DOC 03 The user should be notified of any errors/problems that occurred during the transcoding phase.

DOC 04 Documents should be legible at various resolutions.

DOC 05 Complex path data must be supported

DOC 06 Graphics content should be able to be rendered over raster images

DOC 07 Scalable fonts must be supported

5.5 Use Case, Device Personalization and Extras

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)	X		X	X	X
Additional Keywords				SVG Engine	text/svg

Table 5: Affected Areas for Personalization and Extras

5.5.1 Short Description

Another use of SVG is in device personalization and extras for mobile devices. For example wallpaper can be designed once and be easily traded and displayed across devices of varying display properties. Another possibility is with screensavers. With SVG based screensavers not only can they be displayed on devices of varying display properties but they can be deployed more easily across multiple platforms and traded as easily as ring tones or messages.

5.5.2 Actors

End User – the individual that desires the device personalization and/or extras

Service Provider – the organization that is offering service to the End User

Content Provider – the organization that provides the content to the Service Provider

5.5.2.1 Actor Specific Issues

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must have agreement with Content Provider to provide appropriate content.

Content Provider – Must be willing to provide their content in SVG format.

5.5.2.2 Actor Specific Benefits

End User – The End User is provided with a richer and more fulfilling experience.

Service Provider – People would be willing to pay to customize their device with extras.

5.5.3 Pre-conditions

End User – Must have a mobile device that can display SVG content. This could be a simple viewer or a special SVG enabled client application.

Service Provider – Must have agreement with Content Provider to provide appropriate content.

Content Provider – Must be willing to provide their content in SVG format.

5.5.4 Post-conditions

The End User has the content on their mobile device and gets to enjoy it.

5.5.5 Normal Flow

The following flow shows the interaction:

End User	Service Provider	Content Provider
		1. The content provider has an agreement with the service provider to sell their SVG phone customisations.
2. The user requests to buy a new SVG screensaver from the service provider.		
	3. The service provider receives the request, charges the user's account, and sends the new screensaver to the user via MMS.	
4. The user receives the new screensaver and enjoys it on their phone		

5.5.6 Operational and Quality of Experience Requirements

PER 01 Content should be efficient

PER 02 New content must not break or hide any functionality of the device

PER 03 Complex path data must be supported

PER 04 Scalable fonts must be supported

6. Requirements (Normative)

6.1 High-Level Functional Requirements

6.1.1 General

GEN 1	Rendering hints **may** be allowed (text/shape/image quality, antialiasing).
GEN 2	**Must** be designed to maximize interoperability and efficiency on multiple implementations

6.1.2 Functional

6.1.2.1 Rendering

FUN 1	**MUST** allow zooming, panning, and rotation of content.

6.1.2.2 Shapes

FUN 2	**MUST** support complex path definitions
FUN 3	**MUST** support a set of predefined shapes (i.e. circle, rectangle, etc.)
FUN 4	Shapes **MUST** be able to be filled and/or stroked.

6.1.2.3 Text

FUN 5	**SHOULD** support minimal text layout
FUN 6	**SHOULD** support the definition of user-defined fonts.
FUN 7	**SHOULD** support scalable fonts.

6.1.2.4 Image

FUN 8	**SHOULD** support rendering raster images.
FUN 9	**MUST** allow overlaying graphical objects over a raster/vector image.

6.1.2.5 Animation

FUN 10	**MUST** support animation.
FUN 11	**MUST** allow specification of a certain point in time for static rendering of an animation
FUN 12	**MUST** support declarative animation that is triggered by user interactivity (i.e. mouse click event)

6.1.2.6 Reuse

FUN 13	**MUST** support a way to reuse user-defined content in single file (i.e. define a visual element and instruct it to be rendered multiple times)

6.1.2.7 Interaction

FUN 14	SVG insertion within an XHTML document **MUST** be via the <object> element.

6.1.2.8 Scripting

FUN 15	The event model that is chosen **SHOULD** be compatible with the uDOM specified in the SVGT 1.2 specifications and the APIs defined in JSR-226.[1]

6.1.2.9 Style

FUN 16	The following SVGT CSS properties **MUST** be supported: Font: 'font-family', 'font-size', 'font-style', Text: 'letter-spacing', 'text-decoration', 'word-spacing', Other: 'color', 'display', 'visibility'.

6.2 Requirements Cross-Reference Table

	GEN1	GEN2	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6	FUN7	FUN8	FUN9	FUN10	FUN11	FUN12	FUN13	FUN14	FUN15	FUN16
MAP01			**X**															
MAP02	**X**		**X**														**X**	**X**
MAP03	**X**		**X**														**X**	
MAP04		**X**											**X**		**X**			
MAP05	**X**																**X**	**X**
MAP06				**X**	**X**													**X**
MAP07										**X**	**X**							**X**
MAP08								**X**	**X**									**X**
MAP09							**X**	**X**										**X**
MAP10															**X**			**X**
MAP11											**X**				**X**			

[1] Note that in the JSR-226 spec some of the interfaces in SVGT1.2 have been collapsed, therefore there may be interoperability problems when using scripting intended for SVGT1.2 objects. See Appendix B for details.

	GEN1	GEN2	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6	FUN7	FUN8	FUN9	FUN10	FUN11	FUN12	FUN13	FUN14	FUN15	FUN16
MAP12											X				X			
MAP13			X						X		X							
ANI01		X			X								X		X	X	X	X
ANI02												X		X			X	
ANI03				X	X													X
ANI04										X	X							X
ANI05							X	X	X									X
SUI01		X													X		X	
SUI02																		X
SUI03				X	X													X
SUI04										X	X							X
SUI05							X	X	X									
DOC01		X													X	X	X	X
DOC02	X			X	X	X	X	X	X	X	X						X	X
DOC03																		
DOC04	X		X															X
DOC05				X	X													X
DOC06										X	X							X
DOC07									X									X
PER01		X														X	X	X
PER02																X	X	X
PER03				X	X													X
PER04									X									X

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version V1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-RD-SVGMobileDomain-V1_0	14 May 2004	New Document	First Submission
	22 Sep 2004	Updated Draft	Updated with changes from group suggestions and discussions.
	29 Nov 2004	Updated Draft	Updated with suggestions from formal REQ review.
Candidate Version OMA- RD-SVGMobileDomain-V1_0	12 Jan 2005	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2004-0446R02-SVGMobileDomain-RD-for-TP-Approval

Appendix B. Conformance Considerations (Informative)

There needs to be a strong emphasis on conformance and content authoring guidelines to allow SVG content that is suitable for all mobile devices. Content developers should keep these guidelines in mind to ensure interoperability of SVG across multiple classes of mobile devices with varying degrees of user interface and software sophistication.

In particular, the following areas warrant special consideration :

B.1 The <video> element

This element allows the inclusion of video inside the svg object. Rendering video is inherently a processor-intensive task, therefore excessive use of video while interacting with the SVG object may cause poor performance. Content developers should consider carefully the use cases for video nested inside an SVG object.

In addition, developers are encouraged to make use of the <switch> statement to substitute a static image for devices which do not support SVG video.

B.1.1 Transforming Video

The “transform” attribute is a valid attribute of the <video> element, as well as containers of video such as the <svg>, <g>, <use>, <page>, and <pageSet> elements. It can take on values such as “scale”, “rotate”, “SkewX”, and “SkewY”. Content developers should be aware that performing transformations such as scaling, rotating, and skewing on a video element is resource intensive and may seriously diminish the usability of an SVG object. In addition, some transforms (such as “Skew”) should not be performed on video at all, for obvious usability reasons.

B.2 Transition Effects

The W3C SVG Full specifications provide support for transition effects between different views of an SVG object. Content developers should note that transition effects are not supported in the SVG Tiny specification, and should therefore refrain from making use of transitions in SVG content to ensure interoperability.

B.3 The <handler> element

The <handler> element allows XML events inside the SVG object. It includes mechanisms for XML Event listening, capturing, and bubbling. In order to limit complexity of implementation on mobile devices, the W3C SVG Tiny spec editors have proposed limit support to the bubbling phase only. (i.e. No capturing support) When using the <handler> element, SVG developers should keep these issues in mind to ensure interoperability. It should also be noted that much of what can be achieved with <handler> could also be achieved through script.

B.4 DOM interoperability and JSR-226

Some interfaces present in the SVG specification were “collapsed” in the JSR-226 (Scalable 2D Vector Graphics API for J2ME) specification, and content developers should be aware of the differences in these interfaces when referenced within a Java environment.

Specifically, the following changes were made by the JSR-226 specification has done the following:

1. Moved the methods from the **TraitAccess** and **ElementTraversal** interfaces to the **SVGElement** interfaces, thereby eliminating the TraitAccess and ElementTraversal interfaces altogether.

2. Moved the methods from the **ElementTimeControl** interfaces to the **SVGAnimationElement** interface, thereby eliminating the ElementTimeControl interface altogether.

3. Moved the methods from the **SVGLocatable** interface to the **SVGLocatableElement** interface, thereby eliminating the SVGLocatable interface altogether.

4. Determined that the **EventListenerInitializer** interface is not a must.

As a result of the collapsing of these interfaces, there may be DOM interoperability issues between SVG content written for manipulation through the JSR-226 APIs and content written for consumption outside the Java environment. This should be noted by content developers making use of the affected interfaces.