XML.COM | O'REILLY NETWORK | OREILLY.COM | ONJAVA.COM

Home | Resources | Buyer's Guide | FAQs | Newsletter | Tech Jobs

search

RESOURCE CENTERS

Business

Databases

Graphics

Metadata

Mobile

Programming

Schemas

Style

Web

Web Services

ESSENTIALS
Annotated XML
What is XML?
What is XSLT?
What is XSL-FO?
What is XLink?
What is XML Schema?
What is RDF?
What are Web Services?

Manage Your Account
Forgot Your Password?

FIND
Search
Article Archive
FAQs

COLUMNS
XML-Deviant
Style Matters
XML Q&A
Transforming XML
Perl and XML
XML Endpoints
<taglines>
Sacré SVG

GUIDES
XML Resources
Buyer's Guide
Events Calendar
Standards List
Submissions List

-->

TOOLBOX

Syntax Checker

Print This Article

Server Side SVG

by J. David Eisenberg
February 27, 2002

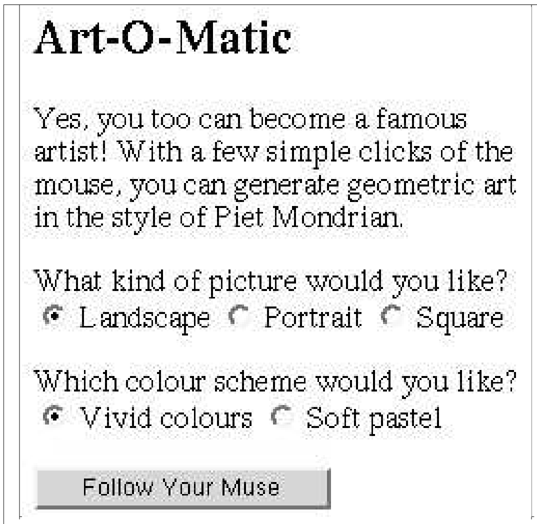
If you've been using SVG or reading XML.com, you probably know about the Adobe SVG Viewer, and you may have heard of the Apache Batik project. Although Batik is most widely known for its SVG viewer component, it's more than that. Batik, according to its web site, is a "Java technology-based toolkit for applications that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as viewing, generation or manipulation."

The Batik viewer application uses the `JSVGCanvas` component, which is a Java class that accepts SVG as input and displays it on screen. In this article, we'll use two of the other Batik components, `SVGGraphics2D`, and the Batik transcoders. The `SVGGraphics2D` class is the inverse of `JSVGCanvas`; you draw into an `SVGGraphics2D` environment using the standard Java two-dimensional graphics methods, and the result is an SVG document. The transcoders take an SVG document as input and produce either JPG or PNG as output.

The context in which we'll use these tools is a servlet that generates geometric art in the style of Piet Mondrian. If the client supports SVG, the servlet will return an SVG document. Otherwise, it will return a JPEG or PNG image, depending upon client support for those image formats.

The Client Side

The web page that we show users will let them choose the orientation and color scheme of their painting. See the screenshot below for an example of how this might look.



Here's the HTML:

```
<html>
<head>
<title>Art-O-Matic</title>
</head>

<body>
<h2>Art-O-Matic</h2>
<p>
Yes, you too can become a famous artist! With a few simple clicks
of the mouse, you can generate geometric art in the style of
Piet Mondrian.
</p>

<form id="artForm"
    action="http://jde:8080/artmaker/servlet/ArtMaker">
<p>
What kind of picture would you like?
<br />
<input type="radio" name="picType" value="landscape"
    checked="checked" /> Landscape
<input type="radio" name="picType" value="portrait" /> Portrait
<input type="radio" name="picType" value="square" /> Square
</p>

<p>
Which color scheme would you like?
<br />
<input type="radio" name="scheme" value="bright"
    checked="checked" /> Vivid colors
<input type="radio" name="scheme" value="pastel" /> Soft pastel
</p>
</body>
</html>
```

Determining whether the client has SVG support or not must be handled on the client side. We'll add this script, taken from a Sun Microsystems technical note, to do this detection.

```
<script type="text/javascript">
<!-- <![CDATA[

var hasSVGSupport = false;  // does client have SVG support?
var useVBMethod = false;    // use VBScript or JavaScript?

/*
```

```
    Internet Explorer returns 0 as the number of MIME types,
    so this code will not be executed by it. This is our indication
    to use VBScript to detect SVG support.
*/
if (navigator.mimeTypes != null
    && navigator.mimeTypes.length > 0)
{
    if (navigator.mimeTypes["image/svg+xml"] != null)
    {
        hasSVGSupport = true;
    }
}
else
{
    useVBMethod = true;
}
// ]]> -->
</script>

<!--
    Visual Basic Script to detect support of Adobe SVG plugin.
    This code is not run on browsers which report they have MIME types,
    and it is also not run by browsers which do not have VBScript support.
-->
<script type="text/vbscript">
On Error Resume Next
If useVBMethod = true Then
    hasSVGSupport = IsObject(CreateObject("Adobe.SVGCtl"))
End If
</script>
```

In order to have the web page send this information back to the server, we'll have to change the `<form>` tag and add a hidden field to our form:

```
<form id="artForm"
    action="http://jde:8080/artmaker/servlet/ArtMaker"
    onsubmit="return setSVGStatus();">

<input type="hidden" name="imgType" value="jpg" />
```

Which requires an additional script to set the hidden field before we send the data to the server:

```
<script type="text/javascript">
<!-- <![CDATA[
function setSVGStatus()
{
    if (hasSVGSupport)
    {
        var theForm = document.getElementById("artForm");
        theForm.imgType.value = "svg";
    }
    return true;
}
// ]]> -->
</script>
```

You may see the source for the entire HTML file.

Pages: 1, 2 Next Page ▶