Adobe® Illustrator® 10 XML Extensions Guide

For Saving as SVG

September 26, 2001

ADOBE SYSTEMS INCORPORATED
Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000
http://www.adobe.com

Contents

1 Introduction

1.1 About this document

This document gives an overview of the XML grammars for the SVG code exported by Adobe® Illustrator® 10 when saving a file in SVG format.

1.2 Adobe Illustrator 10 XML Grammars

The following are the various XML grammars that are exported from Illustrator 10 in various circumstances:

- Adobe Illustrator 10 XML Elements and Attributes
- Extensibility XML Elements and Attributes
- Text Flows XML Grammar
- Graphing XML Grammar
- Image Replacement XML Grammar
- Save For Web (SFW) XML Grammar
- Variable Bindings and Variable Definitions XML Grammar

1.3 Overview of SVG Generated by Adobe Illustrator 10

The following two examples provide an overview of the SVG files generated by Adobe Illustrator 10. The first example shows the structure of an SVG file when the user turns off the checkbox “Preserve Illustrator Editing Capabilities” on the *SaveAs* SVG Options dialog. The second example shows the structure when the checkbox is turned on.

In the first example below, the user has turned off the “Preserve Illustrator Editing Capabilities” checkbox. The first two lines contain the `<?xml...?>` processing instruction and a comment indicating that the file was generated by Adobe Illustrator 10.

The `DOCTYPE` declaration starts on the third line. The `DOCTYPE` specifies that this SVG file is compatible with SVG 1.0 Recommendation’s DTD and includes a series of entity declarations for various namespace URIs, such as “`ns_svg`”, which represents the namespace URI of SVG 1.0 (“http://www.w3.org/2000/svg”). After the `DOCTYPE` declaration comes the root `<svg>` element.

If the user specifies that fonts are to be embedded, then the SVG file will contain a `<style>` element, inside of which there will be an `@font-face` rule for each embedded font. The

embedded font data will be specified using the 'data:' protocol to specify the value for the 'src:' descriptor.

If the user has requested *Include Extended Syntax for Variable Data*, *Include Slicing Data*, or *Include File Info* (these options are available on the *Advanced options* dialog), then a <metadata> element will be included.

After the <metadata> comes the actual graphics for the document. Each Illustrator layer is contained within a separate <g> element.

Note:

The namespace "http://ns.adobe.com/GenericCustomNamespace/1.0/" is a default namespace URI that can be used for any elements within the sample data sets which correspond to variable names. For example, if you define a variable named "v1" in Illustrator, then the sample data sets will include an element named "v1", as in <v1>...</v1>. In order to conform to the W3C XML Namespace specification and to the SVG specification, these elements need to be a defined namespace other than the SVG namespace. Therefore, Illustrator exports its files with all variables in the "http://ns.adobe.com/GenericCustomNamespace/1.0/" namespace.

EXAMPLE 1.1

```
<?xml version="1.0" ...?>
<!-- Generator: Adobe Illustrator 10.0 ...  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
      "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd" [
  <!ENTITY ns_svg "http://www.w3.org/2000/svg">
  <!ENTITY ns_ai "http://ns.adobe.com/AdobeIllustrator/10.0/">
  <!ENTITY ns_xlink "http://www.w3.org/1999/xlink">
  ...
]>
<svg xmlns="&ns_svg;" xmlns:xlink="&ns_xlink;" xmlns:i="&ns_ai;" ...>

  <!-- Embedded fonts are defined in @font-face rules -->
  <style type="text/css">
      <![CDATA[
           @font-face{font-family:'Myriad-Roman';
                      src:url("data:;base64,---embedded font data---")}
      ]]>
    </style>

  <metadata>
    <!--  Various types of document global data go here:
           * Variable definitions and corresponding sample data
           * Save-for-web settings
           * General metadata from "File Info" dialog
    -->
  </metadata>
```

```
  <g id="Layer1" ...>
    ...graphics for layer 1 goes here...
  </g>
  <g id="Layer2" ...>
    ...graphics for layer 2 goes here...
  </g>
  <!-- more layers -->

</svg>
```

In the second example below, the user has turned on the *Preserve Illustrator Editing Capabilities* checkbox. This checkbox causes a large block of binary data to be added to the end of the SVG file inside a <pgf> element.. Also, a <switch> element will bracket all of the graphics in the document. The <switch> is set up so that when Illustrator itself opens the file, it will recognize the "requiredExtensions" value on the <foreignObject>, will import using the <foreignObject> and the referenced <pgf> data and will ignore the SVG version of the graphics. Other SVG implementations, on the other hand, including the Adobe SVG Viewer, will ignore the <foreignObject> and the <pgf> and will instead render using the SVG version of the contents.

NOTE: If the SVG contents of the file below are modified (e.g., by hand-editing in a text editor), the modifications will be ignored when Illustrator opens the file because the <pgf>, if present, takes precedence. For more on the <pgf> element, refer to Chapter 8, "Adobe Illustrator 10 XML Elements and Attributes."

EXAMPLE 1.2

```
<?xml version="1.0" ...?>
<!-- Generator: Adobe Illustrator 10.0 ...  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
      "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd" [
 ...Same as previous example  ...
]>
<svg xmlns="&ns_svg;" xmlns:xlink="&ns_xlink;" xmlns:i="&ns_ai;" ...>

  <style type="text/css">
    ...same as previous example...
  </style>

  <metadata>
    ...same as previous example...
  </metadata>

  <switch>

    <!-- Reference to PGF data, which is at end of file. If PGF data
         is present, Adobe Illustrator 10 will read the PGF data
         and ignore the corresponding SVG data. -->
    <foreignObject requiredExtensions="&ns_ai;" x="0" y="0" width="1" height="1">
      <i:pgfRef  xlink:href="#adobe_illustrator_pgf">
      </i:pgfRef>
```

```
    </foreignObject>

    <!-- SVG viewers will skip past the <foreignObject> above
         and will render the SVG data below instead. -->
    <g i:extraneous="self">

      <g id="Layer1" ...>
        ...graphics for layer 1 goes here...
      </g>
      <g id="Layer2" ...>
        ...graphics for layer 2 goes here...
      </g>
      <!-- more layers -->
    </g>
  </switch>
  <i:pgf id="adobe_illustrator_pgf">
    <![CDATA[
      ...stream of base64-encoded binary goes here...
    ]]>
  </i:pgf>

</svg>
```

1.4 Sample SVG

The following is a sample SVG file from Illustrator 10. The file consists of a single text element which is positioned inside of a triangle (expressed as a path element). The *SaveAs* SVG Options (including Advanced options) specified the following:

On the main Save As SVG Options dialog:

- Subsetting: Only glyphs used
- Embed fonts
- Embed images (but there are no images in this example)
- Preserve Illustrator Editing Capability

On the Advanced options dialog:

- Optimize for Adobe SVG Viewer (but there are no optimizations in this file)
- Include Extended Syntax for Variable Data
- Include Slicing Data
- Include File Info

EXAMPLE 1.3 Sample SVG

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!-- Generator: Adobe Illustrator 10.0, SVG Export Plug-In . SVG Version: 3.0.0 Build 41)
-->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"    "http://www.w3.org/TR/2001/PR-SVG-
20010719/DTD/svg10.dtd" [
      <!ENTITY ns_flows "http://ns.adobe.com/Flows/1.0/">
      <!ENTITY ns_svg "http://www.w3.org/2000/svg">
      <!ENTITY ns_extend "http://ns.adobe.com/Extensibility/1.0/">
      <!ENTITY ns_ai "http://ns.adobe.com/AdobeIllustrator/10.0/">
      <!ENTITY ns_graphs "http://ns.adobe.com/Graphs/1.0/">
      <!ENTITY ns_vars "http://ns.adobe.com/Variables/1.0/">
      <!ENTITY ns_imrep "http://ns.adobe.com/ImageReplacement/1.0/">
      <!ENTITY ns_sfw "http://ns.adobe.com/SaveForWeb/1.0/">
      <!ENTITY ns_custom "http://ns.adobe.com/GenericCustomNamespace/1.0/">
      <!ENTITY ns_adobe_xpath "http://ns.adobe.com/XPath/1.0/">
      <!ENTITY ns_xlink "http://www.w3.org/1999/xlink">
]>
<svg  xmlns="&ns_svg;" xmlns:xlink="&ns_xlink;" xmlns:x="&ns_extend;" xmlns:i="&ns_ai;"
      i:viewOrigin="211.3462 507.0288" i:rulerOrigin="0 0" i:pageBounds="0 792 612 0"
      xmlns:a="http://ns.adobe.com/AdobeSVGViewerExtensions/3.0/"
      width="153.414" height="77.251" viewBox="0 0 153.414 77.251"
      overflow="visible" enable-background="new 0 0 153.414 77.251" xml:space="preserve">

      <!-- The embedded font is included with the 'src' descriptor for the @font-face rule
-->
      <style type="text/css">
      <![CDATA[
@font-face{font-family:'GenericFont';src:url("data:;base64,\
T1RUTwADACAAAQAQQ0ZGIH/OWx8AAAA8AAAEY0dQT1OwTb9OAAAFDAAAAGBjbWFwAlQCSQAABKAA\

   ...Many more lines of base64-encoded binary...

AA8ABwAPAAgAGQAJAB4AAQABAAo=")}
      ]]>
      </style>

      <metadata>

          <!-- Information about slices (created via slice tool) and
          Save For Web options (defined in Save For Web dialog). -->
          <sfw xmlns="&ns_sfw;">
              <slices>
                  <slice  x="240" y="436" name="" type="1" width="97" height="15"
sliceID="1"
                      groupID="331629544" url="" target="" message="" altTag=""
                      cellTextIsHTML="false" cellText="" horzAlign="1" vertAlign="1"
                      background="none">
                  </slice>
              </slices>
              <optimizationSettings>
                  <targetSettings  fileFormat="GIFFormat" targetSettingsID="331629544">
                         <GIFFormat  transparency="true" includeCaption="false"
                             interlaced="false" noMatteColor="false"
```

```
                                matteColor="#FFFFFF" autoReduce="false"
                            rolloverMasterPalette="false" webShiftPercent="0"
                                numColors="256" lossy="0" ditherAlgorithm="diffusion"
                                ditherPercent="100" reductionAlgorithm="selective">
                            </GIFFormat>
                        </targetSettings>
                    </optimizationSettings>
                    <sliceSourceBounds  x="211.346" y="429.778" width="153.414"
                            height="77.251" bottomLeftOrigin="true">
                    </sliceSourceBounds>
                </sfw>

                <!-- Variable definitions and associated sample data (defined in Variables
palette) -->
                <variableSets  xmlns="&ns_vars;">
                    <variableSet  varSetName="binding1" locked="none">
                        <variables>
                            <variable  varName="Variable1" trait="textcontent"
category="&ns_flows;"
                                docRef="id('XMLID_1_')"></variable>
                        </variables>
                        <v:sampleDataSets  xmlns="&ns_custom;" xmlns:v="&ns_vars;"
                                i:activeDataSet="Data Set 2">
                            <v:sampleDataSet  dataSetName="Data Set 1">
                                <Variable1>
                                    <p>Sample text 1</p>
                                </Variable1>
                            </v:sampleDataSet>
                            <v:sampleDataSet  dataSetName="Data Set 2">
                                <Variable1>
                                    <p>Here is sample 2</p>
                                </Variable1>
                            </v:sampleDataSet>
                        </v:sampleDataSets>
                    </variableSet>
                </variableSets>

            <!-- General metadta from Illustrator's File Info dialog, output as RDF -->
            <svgXAP><?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d' bytes='2140'?><?adobe-xap-
filters esc="CR"?>
              <x:xapmeta xmlns:x='adobe:ns:meta/'>
              <rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
               xmlns:iX='http://ns.adobe.com/iX/1.0/'>

               <rdf:Description about=''
                xmlns='http://ns.adobe.com/pdf/1.3/'
                xmlns:pdf='http://ns.adobe.com/pdf/1.3/'>
                <pdf:CreationDate>2001-07-27T00:57:54Z</pdf:CreationDate>
                <pdf:ModDate>2001-07-28T18:39:41Z</pdf:ModDate>
                <pdf:Author>Me</pdf:Author>
               </rdf:Description>
```

```
        <rdf:Description about=''
         xmlns='http://ns.adobe.com/xap/1.0/'
         xmlns:xap='http://ns.adobe.com/xap/1.0/'>
         <xap:CreateDate>2001-07-27T00:57:54Z</xap:CreateDate>
         <xap:ModifyDate>2001-07-28T21:59:27Z</xap:ModifyDate>
         <xap:CreatorTool>Adobe Illustrator 10.0</xap:CreatorTool>
         <xap:Author>Me</xap:Author>
         <xap:Title>Generic AI File</xap:Title>
         <xap:Description>Contains a bit of </xap:Description>
         <xap:Keywords>
          <rdf:Bag>
           <rdf:li>generic</rdf:li>
           <rdf:li>illustrator</rdf:li>
           <rdf:li>file</rdf:li>
          </rdf:Bag>
         </xap:Keywords>
        </rdf:Description>

        <!-- Additional RDF information removed from this sample for brevity -->

      </rdf:RDF>
      </x:xapmeta>
      <?xpacket end='r'?>
    </svgXAP>
      </metadata>

      <switch>

          <foreignObject requiredExtensions="&ns_ai;" x="0" y="0" width="1"
                height="1">
              <i:pgfRef  xlink:href="#adobe_illustrator_pgf">
              </i:pgfRef>
          </foreignObject>

          <g i:extraneous="self">
              <g id="Layer1" i:layer="yes" i:dimmedPercent="50" i:color="#4F008000FFFF"
stroke="#000000">
                  <path fill="#FF0000" stroke-width="4"
d="M4.817,74.9L76.881,2.835l71.725,72.405L4.817,74.9z"/>
              </g>
              <g id="Layer2" i:layer="yes" i:dimmedPercent="50" i:color="#FFFF4F004F00"
stroke="#000000">
                  <switch id="XMLID_1_" i:objectNS="&ns_flows;" i:objectType="pointText"
transform="matrix(1 0 0 1 30.5044 65.4648)">
                      <foreignObject requiredExtensions="&ns_flows;" x="0" y="0" width="1"
height="1" overflow="visible" stroke="none">
                          <flowDef  xmlns="&ns_flows;">
                              <region>
                                  <path d="M30.504,65.465"/>
                              </region>
                              <flow  xmlns="&ns_flows;">
```

```
                                    <p><span  font-family="'GenericFont'" font-size="12">Here
is sample 2</span></p>
                                </flow>
                            </flowDef>
                            <x:targetRef  xlink:href="#XMLID_2_" />
                        </foreignObject>
                        <g id="XMLID_2_">
                            <text><tspan stroke="none" font-family="'GenericFont'" font-
size="12">Here is sample 2</tspan></text>
                        </g>
                    </switch>
                    <path id="_x003C_Slice_x003E_" fill="none" stroke="none"
d="M28.654,71.029v-15h97v15H28.654z"/>
                </g>
            </g>
        </switch>

        <i:pgf  id="adobe_illustrator_pgf">
            <![CDATA[
            eJzsvWtzJMeRIPidZv0f6j6MmXQ3RGe8I3Vja1ZVAOa0hhFlJLUzezNrNAgNkb2D7ub2Q1rurz9/
RnhkZgHoZmvFOQNSTSEdkZGRER7+do+/+z9+/82X+xdv/nj7ZTibds+++Lu/O769X7/5u1vdgTe

   ...Many more lines of base64-encoded binary...

TxZJwdS/oU8oU8ywtbV/ctB493/9zOqU
            ]]>
        </i:pgf>
</svg>
```

2 Variable Bindings and Variable Definitions XML Grammar

2.1 About This Chapter

This chapter describes the XML grammar for variable bindings and variable definitions files.

2.2 About Variable Bindings and Variable Definitions

Variable bindings contain an XML representation of the contents of the variable palette. The variable bindings information is included in the SVG file exported by Adobe Illustrator when the user chooses the “Include Extended Syntax for Variable Data” option. Within the SVG file, the variable bindings information is saved in elements which are in the variables namespace.

Variable definitions are what is saved when a user executes a *Capture Data Set* command from the variables palette. The variable definitions represent a subset of the XML used for variable bindings.

Variable bindings and variable definitions have separate but overlapping DTDs and use the same XML namespace.

2.3 Namespace

The following is the namespace URI for variable bindings and variable definitions:

```
http://ns.adobe.com/Variables/1.0/
```

2.4 DTD for Variable Definitions and Variable Bindings

The DTD for variable and binding definitions files is shown below.

The presence or absence of a `docRef` indicates whether the DTD is for variable definitions or for variable bindings; if the `docRef` is present, it is a bindings file, and if it’s absent, it is a definitions file.

```
<!-- VariableBindings.dtd (version 1.0) - 20010915 -->

<!-- This DTD summarizes the elements and attributes in the
     variable bindings namespace exported from Adobe Illustrator 10.0..
     The namespace URI is "http://ns.adobe.com/Variables/1.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-VARS (Namespace prefix for elements named var*
```

```
                    in this namespace, such as "v:")
        XMLNS-DECLARE-VARS (xmlns[:prefix] attribute declaration)
        NSPREFIX-DATA (Namespace prefix for elements named sampleData*
                    in this namespace, such as "v:")
        XMLNS-DECLARE-DATA (xmlns[:prefix] attribute declaration)
        XMLNS-DECLARE-CUSTOM (usually, actual data will be in a custom,
                            user-defined namespace)
        NSPREFIX-AI (the 'activeDataSet' attribute in the Illustrator10
                    namespace needs this)
        and then include this DTD via reference. -->

<!-- ================================================
                Data type declarations
     ================================================ -->

<!ENTITY % name "CDATA" >
  <!-- Defined to match the 'name' production in the XML 1.0 spec -->
<!ENTITY % docref "CDATA" >
  <!-- Defined to be the subset of XPath supported by Pipestone -->
<!ENTITY % uri "CDATA" >
  <!-- Same constraints as href attribute in XLink. -->


<!-- ================================================
                Element declarations
     ================================================ -->

<!ENTITY % variableSetsElement "%NSPREFIX-VARS;variableSets" >
<!ENTITY % variableSetElement "%NSPREFIX-VARS;variableSet" >
<!ENTITY % variablesElement "%NSPREFIX-VARS;variables" >
<!ENTITY % variableElement "%NSPREFIX-VARS;variable" >
<!ENTITY % sampleDataSetsElement "%NSPREFIX-DATA;sampleDataSets" >
<!ENTITY % sampleDataSetElement "%NSPREFIX-DATA;sampleDataSet" >

<!ELEMENT %variableSetsElement; (%variableSetElement;)* >
<!ATTLIST %variableSetsElement;
  %XMLNS-DECLARE-VARS; >

<!ELEMENT %variableSetElement; ((%variablesElement;)*,(%sampleDataSetsElement;))* >
<!ATTLIST %variableSetElement;
  %XMLNS-DECLARE-VARS;
  varSetName %name; #IMPLIED
  locked (variablesAndTypes|none) 'none' >

<!ELEMENT %variablesElement; (%variableElement;)* >
<!ATTLIST %variablesElement;
  %XMLNS-DECLARE-VARS; >

<!ELEMENT %variableElement; EMPTY >
<!ATTLIST %variableElement;
  %XMLNS-DECLARE-VARS;
  varName %name; #REQUIRED
```

```
  docRef %docref; #IMPLIED
  category %uri; #IMPLIED
  trait %name; #IMPLIED >

<!ELEMENT %sampleDataSetsElement; (%sampleDataSetElement;)* >
<!ATTLIST %sampleDataSetsElement;
  %XMLNS-DECLARE-DATA;
  %XMLNS-DECLARE-CUSTOM;
  %IllustratorSampleDataSetsAttributes;
  >

<!ELEMENT %sampleDataSetElement; ANY >
<!ATTLIST %sampleDataSetElement;
  %XMLNS-DECLARE-DATA;
  %XMLNS-DECLARE-CUSTOM;
  dataSetName %name; #IMPLIED
  >
```

2.5 Example

The following is sample XML for the variable bindings XML grammar.

```
<?xml version="1.0" ...?>
<!-- Generator: Adobe Illustrator 10.0 ...  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
      "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd" [
  <!ENTITY ns_svg "http://www.w3.org/2000/svg">
  <!ENTITY ns_ai "http://ns.adobe.com/AdobeIllustrator/10.0/">
  <!ENTITY ns_xlink "http://www.w3.org/1999/xlink">
  <!ENTITY ns_flows "http://ns.adobe.com/Flows/1.0/">
  <!ENTITY ns_vars "http://ns.adobe.com/Variables/1.0/">
  <!ENTITY ns_custom "http://ns.adobe.com/GenericCustomNamespace/1.0/">
  ...
]>
<svg xmlns="&ns_svg;" xmlns:xlink="&ns_xlink;" xmlns:i="&ns_ai;" ...>
  <metadata>

    <!-- Other document-global data goes here, also -->

    <!-- These are the variable bindings  -->
    <variableSets xmlns="&ns_vars;">
      <variableSet varSetName="binding1">
        <variables>
          <variable varName="variable1" docRef="id('Layer_1')"
                    category="&ns_vars;" trait="visibility"/>
          <variable varName="variable2" docRef="id('Text_1')"
                    category="&ns_flows;" trait="textcontent"/>
        </variables>
      </variableSet>
      <v:sampleDataSets xmlns:v="&ns_vars;" xmlns="&ns_custom;">
        <v:sampleDataSet dataSetName="Mock Data Set 1">
```

```
            <variable1>false</variable1>
            <variable2>
              <flow  xmlns="&ns_flows;">
                <p>
                  <span>Text for sample data set 1.</span>
                </p>
              </flow>
            </variable2>
          </v:sampleDataSet>
          <v:sampleDataSet dataSetName="Mock Data Set 2">
            <variable1>true</variable1>
            <variable2>
              <flow  xmlns="&ns_flows;">
                <p>
                  <span>Text for sample data set 2.</span>
                </p>
              </flow>
            </variable2>
          </v:sampleDataSet>
        </v:sampleDataSets>
      </variableSets>
    </metadata>

    <!-- Rendered document tree goes here -->

  </svg>
```

The `trait` attribute would indicate the class of variable and would correspond to the icon in the far left column of the variables palette. For example, one type of `trait` would be `graphdata`.

For the four classes of variables in Illustrator 10:

```
category="http://ns.adobe.com/Graphs/1.0/" trait="graphdata"
category="http://ns.adobe.com/Flows/1.0/" trait="textcontent"
category="http://ns.adobe.com/Variables/1.0/" trait="visibility"
category="http://ns.adobe.com/Variables/1.0/" trait="fileref"
```

When `trait="visibility"`, AlterCast maps `visibility=true|false` to `display=inline|none`.

When `trait="fileref"`, AlterCast replaces the image with the target object.

3

Text Flows XML Grammar

3.1 About This Chapter

This chapter describes the XML grammar and property definitions for text flows which will be embedded in some SVG files exported by Adobe Illustrator 10.

3.2 Namespace

The following is the namespace URI for text flows:

```
http://ns.adobe.com/Flows/1.0/
```

3.3 DTD

The following is the DTD for the text flow namespace:

```
<!-- Flows.dtd (version 1.0) - 20010915 -->

<!-- This DTD summarizes the elements and attributes in the
     Flows namespace exported from Adobe Illustrator 10.0..
     The namespace URI is "http://ns.adobe.com/Flows/1.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-FLOWS (Namespace prefix for elements named var*
          in this namespace, such as "v:")
       XMLNS-DECLARE-FLOWS (xmlns[:prefix] attribute declaration)
       XMLNS-DECLARE-EMBEDDED-SVG (xmlns[:prefix] attribute declaration)
       NSPREFIX-EMBEDDED-SVG (Namespace prefix for elements in SVG namespace
embedded in Flow ns elements)
       EMBEDDED-PATH-ELEMENT ('region' and 'wrap' require svg:path element child) and
then include this DTD via reference. -->

<!-- Data types -->
<!ENTITY % BlendingMode
"(normal|multiply|screen|difference|darken|lighten|colorDodge|
colorBurn|exclusion|hardLight|overlay|softLight|luminosity|hue|
saturation|color|compatibleOverprint|inherit)">

<!ENTITY % BooleanInherit "(true|false|inherit)" >

<!ENTITY % ClipFillRule "(nonzero | evenodd | inherit)">
    <!-- 'clip-rule' or fill-rule property/attribute value -->

<!ENTITY % EmsLength "CDATA">
       <!-- An SVG length value that must be given in ems -->
```

```
<!ENTITY % FontFamilyValue "CDATA">
    <!-- 'font-family' property/attribute value (i.e., list of fonts) -->

<!ENTITY % FontSizeValue "CDATA">
    <!-- 'font-size' property/attribute value -->

<!ENTITY % Matrix "CDATA">
        <!-- a list of 6 <number>'s separated by white space defining a
             transformation matrix -->

<!ENTITY % Multiplier "CDATA">
        <!-- An SVG <number>  used to multiply some other value -->

<!ENTITY % NumberZeroToOne "CDATA">
        <!-- A number in the range [0, 1] -->

<!ENTITY % NonNegativeInteger "CDATA">
        <!-- An integer greater than or equal to zero -->

<!ENTITY % OpacityValue "CDATA">
    <!-- opacity value (e.g., <number>) -->

<!ENTITY % Paint "CDATA">
    <!-- a 'fill' or 'stroke' property/attribute value: <paint> -->

<!ENTITY % SpacingValue "CDATA">
    <!-- 'letter-spacing' or 'word-spacing' property/attribute value (e.g., normal |
<length>) -->

<!ENTITY % StrokeDashArrayValue "CDATA">
    <!-- 'stroke-dasharray' property/attribute value (e.g., 'none', list of <number>s)
-->

<!ENTITY % StrokeDashOffsetValue "CDATA">
    <!-- 'stroke-dashoffset' property/attribute value (e.g., 'none', <legnth>) -->

<!ENTITY % StrokeMiterLimitValue "CDATA">
    <!-- 'stroke-miterlimit' property/attribute value (e.g., <number>) -->

<!ENTITY % StrokeWidthValue "CDATA">
    <!-- 'stroke-width' property/attribute value (e.g., <length>) -->

<!ENTITY % URI "CDATA">
    <!-- a Uniform Resource Identifier -->

<!ENTITY % UserLength "CDATA">
        <!-- An SVG length value that must be given in user units -->


<!-- Note that due to the limitations of DTDs, the DTD cannot express      the needed
constraints. Additional constraints on the grammar are      that a <flowDef> must have:
```

```
    (1) exactly one child that is either a flow or a flowRef
        (note: flowRef is not supported at this time)
    (2) at least one child that it either a region or a regionRef
        (any number of region or regionRef elements can be present
         so long as there is at least one that is either a region
         or a regionRef)
        (note: regionRef is not supported at this time)
    (3) any number of wrap or wrapRef children
        (note: wrapRef is not supported at this time)
  -->

<!ELEMENT flowDef (flow|flowRef|region|regionRef|wrap|wrapRef)* >
<!ATTLIST flowDef
   xmlns CDATA #FIXED "http://ns.adobe.com/Flows/1.0/"
   xmlns:svg CDATA #FIXED "http://www.w3.org/2000/svg" >

<!-- Contains a flow of objects. Right now, only paragraphs of text are supported. -->
<!ELEMENT flow (p*) >
<!ATTLIST flow
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
>

<!-- Contains a paragraph, similar to a <p> in HTML. -->
<!ELEMENT p (#PCDATA|span)* >
<!ATTLIST p
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  %Properties-Graphics;
  %Properties-TextAndFont;
>

<!-- Contains a run of text, similar to a <span> in HTML. -->
<!ELEMENT span (#PCDATA|span)* >
<!ATTLIST span
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  %Properties-Graphics;
  %Properties-TextAndFont;
>

<!-- [ Defined for possible future use.
     Not used by Adobe Illustrator 10 or Adobe AlterCast 1. ]
     This element points to a <flow> element in this document
     or somewhere on the general Web using XPointer syntax. -->
<!ELEMENT flowRef EMPTY >
<!ATTLIST flowRef
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  %XMLNS-DECLARE-XLINK;
  %xlinkShowAttribute; (other) #FIXED 'other'
  %xlinkActuateAttribute; (onLoad) #FIXED 'onLoad'
```

```
  %xlinkHrefAttribute; %URI; #REQUIRED >

<!-- This element must contain a single SVG <path>.
     Defines a region into which a flow should be poured.
     The flow is poured into the first region/regionRef, then overflows to the second,
etc. -->
<!ELEMENT region (%EMBEDDED-PATH-ELEMENT;) >
<!ATTLIST region
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  textMatrix %Matrix; #IMPLIED
  writing-mode (lr | tb | inherit) #IMPLIED
  %XMLNS-DECLARE-EMBEDDED-SVG;  >

<!-- [ Defined for possible future use.
     Not used by Adobe Illustrator 10 or Adobe AlterCast 1. ]
     This element points to an SVG <path> element in this document
     or somewhere on the general Web using XPointer syntax.
     The referenced element defines a region into which a flow should be poured. -->
<!ELEMENT regionRef EMPTY >
<!ATTLIST regionRef
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  textMatrix %Matrix; #IMPLIED
  writing-mode (lr | tb | inherit) #IMPLIED
  %XMLNS-DECLARE-XLINK;
  %xlinkShowAttribute; (other) #FIXED 'other'
  %xlinkActuateAttribute; (onLoad) #FIXED 'onLoad'
  %xlinkHrefAttribute; %URI; #REQUIRED >

<!-- This element must contain a single SVG <path>.
     Defines a region which a flow must flow around.. -->
<!ELEMENT wrap (%EMBEDDED-PATH-ELEMENT;) >
<!ATTLIST wrap
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  %XMLNS-DECLARE-EMBEDDED-SVG;  >

<!-- [ Defined for possible future use.
     Not used by Adobe Illustrator 10 or Adobe AlterCast 1. ]
     This element points to an SVG <path> element in this document
     or somewhere on the general Web using XPointer syntax.
     The referenced element defines a region which a flow must flow around. -->
<!ELEMENT wrapRef EMPTY >
<!ATTLIST wrapRef
  %XMLNS-DECLARE-FLOWS;
  id ID #IMPLIED
  %XMLNS-DECLARE-XLINK;
  %xlinkShowAttribute; (other) #FIXED 'other'
  %xlinkActuateAttribute; (onLoad) #FIXED 'onLoad'
  %xlinkHrefAttribute; %URI; #REQUIRED >
```

Note:

The `i:objectType` attribute on the <switch> statement currently has three types: `pointText`, `areaText` and `pathText`.

3.4 Example

The following is sample XML for the text flow XML grammar embedded in an SVG file exported by Adobe Illustrator 10.

```
<?xml version="1.0"?>
<!-- Generator: Adobe Illustrator 10.0 ... -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"    "http://www.w3.org/TR/2001/PR-SVG-
20010719/DTD/svg10.dtd" [
  <!ENTITY ns_flows "http://ns.adobe.com/Flows/1.0/">
  <!ENTITY ns_svg "http://www.w3.org/2000/svg">
  <!ENTITY ns_extend "http://ns.adobe.com/Extensibility/1.0/">
  <!ENTITY ns_ai "http://ns.adobe.com/AdobeIllustrator/10.0/">
  <!ENTITY ns_xlink "http://www.w3.org/1999/xlink">
]>
<svg  xmlnsk="&ns_svg;" xmlns:x="&ns_extend;" xmlns:i="&ns_ai;" ...>

  <!-- Not shown: document resources and metadata -->

  <g id="Layer_1" ...>

    <switch i:objectNS="&ns_flows;" i:objectType="pointText"
            transform="matrix(1 0 0 1 37.0234 282.186)" font-size="12">
      <foreignObject requiredExtensions="&ns_flows;"
              x="0" y="0" width="1" height="1" overflow="visible">
        <flowDef  xmlns="&ns_flows;">
          <region>
            <path d="M37.023,282.186"/>
          </region>
          <flow  xmlns="&ns_flows;">
            <p><span  font-family="'Myriad-Roman'" font-size="12">Point text</span></p>
          </flow>
        </flowDef>
        <x:targetRef  xlink:href="#XMLID_1_" />
      </foreignObject>
      <g id="XMLID_1_">
        <text><tspan stroke="none" font-family="'Myriad-Roman'" font-size="12">Point
text</tspan></text>
      </g>
    </switch>
    <switch i:objectNS="&ns_flows;" i:objectType="pathText"
            transform="matrix(0.4845 -0.8748 0.8748 0.4845 42.6367 363.6611)" font-
size="12">
      <foreignObject requiredExtensions="&ns_flows;"
              x="0" y="0" width="1" height="1" overflow="visible">
```

```
        <flowDef  xmlns="&ns_flows;">
          <region>
            <path id="XMLID_2_" fill="none" d="M40.512,371.721c1.163-19.768,45.349-
60.464,73.256-39.534"/>
          </region>
          <flow  xmlns="&ns_flows;">
            <p>Path text</p>
          </flow>
        </flowDef>
        <x:targetRef  xlink:href="#XMLID_2_" />
      </foreignObject>
      <g id="XMLID_2_">
        <path fill="none" stroke="none" d="M40.512,371.721c1.163-19.768,45.349-
60.464,73.256-39.534"/>
        <text>
<tspan y="0" stroke="none">P</tspan>
<tspan x="5.846" y="0.001" stroke="none" rotate="5.993">a</tspan>
<tspan x="11.397" y="0.613" stroke="none" rotate="10.047">t</tspan>
<tspan x="15.178" y="1.236" stroke="none" rotate="13.969">h</tspan>
<tspan x="21.513" y="2.868" stroke="none" rotate="17.124"> </tspan>
<tspan x="23.875" y="3.578" stroke="none" rotate="19.305">t</tspan>
<tspan x="27.456" y="4.797" stroke="none" rotate="22.597">e</tspan>
<tspan x="32.841" y="7.037" stroke="none" rotate="26.579">x</tspan>
<tspan x="37.82" y="9.555" stroke="none" rotate="30.228">t</tspan>
        </text>
      </g>
    </switch>
    <switch i:objectNS="&ns_flows;" i:objectType="areaText"
            transform="matrix(1 0 0 1 42.8374 416.3164)" font-size="12">
      <foreignObject requiredExtensions="&ns_flows;"
              x="0" y="0" width="1" height="1" overflow="visible">
        <flowDef  xmlns="&ns_flows;">
          <region>
            <path id="XMLID_3_" fill="none" d="M120.744,457.768H42.837v-
52.326h77.907V457.768z"/>
          </region>
          <flow  xmlns="&ns_flows;">
            <p>Text within a rectangle with wrapping.</p>
          </flow>
        </flowDef>
        <x:targetRef  xlink:href="#XMLID_3_" />
      </foreignObject>
      <g id="XMLID_3_">
        <path fill="none" stroke="none" d="M120.744,457.768H42.837v-
52.326h77.907V457.768z"/>
        <text>
<tspan stroke="none">Text within a </tspan>
<tspan y="14.5" stroke="none">rectangle with </tspan>
<tspan y="29" stroke="none">wrapping.</tspan>
        </text>
      </g>
    </switch>
  </g>
</svg>
```

4 Save For Web (SFW) XML Grammar

4.1 About This Chapter

This chapter describes the XML grammar for the *Save For Web* (SFW) option in Adobe Illustrator 10.

4.2 About SFW

SFW is a ‘File’ menu command, and one of the key uses of this command is to set per-slice optimization settings. The most recent settings need to be saved in the SVG file exported by Illustrator so that AlterCast can apply these settings when it generates image output. Current SFW options can be saved from the SFW dialog into a file for later re-use with other documents.

The XML grammar shown below serves the needs of embedding SFW information in the SVG file exported by Adobe Illustrator 10 that contains a snapshot of SFW settings.

4.3 Namespace

There are separate namespaces for SFW and for the optimization settings that are stored as part of the SFW data.

The following is the namespace URI for SFW:

```
http://ns.adobe.com/SaveForWeb/1.0/
```

The following is the namespace URI for optimization settings:

```
http://ns.adobe.com/webOptimization/2.0/
```

4.4 DTD

The DTD for the Save For Web option is shown below.

If the `sfwElement` has optimization information with `targetSettingsID=0`, then AlterCast will use those settings for its output settings.

```
<!-- SaveForWeb.dtd (version 1.0) 20011008 -->

<!-- This DTD summarizes the Save For Web namespace
     exported from Adobe Illustrator 10.0 and supported by
     AlterCast 1.0.
```

```
      The namespace URI is "http://ns.adobe.com/SaveForWeb/1.0/"
      This DTD is formulated under the expectation that a parent
      DTD will set up the following entities:
        NSPREFIX-SFW (Namespace prefix for this namespace, such as "sfw:")
        XMLNS-DECLARE-SFW (xmlns[:prefix] attribute declaration)
      and then include this DTD via reference. -->

<!ENTITY % sfwElement "%NSPREFIX-SFW;sfw" >
<!ENTITY % slicesElement "%NSPREFIX-SFW;slices" >
<!ENTITY % sliceElement "%NSPREFIX-SFW;slice" >
<!ENTITY % cellTextElement "%NSPREFIX-SFW;cellText" >
<!ENTITY % resamplingParamsElement "%NSPREFIX-SFW;resamplingParams" >
<!ENTITY % outputSettingsElement "%NSPREFIX-SFW;outputSettings" >
<!ENTITY % htmlOutputSettingsElement "%NSPREFIX-SFW;htmlOutputSettings" >
<!ENTITY % backgroundOutputSettingsElement "%NSPREFIX-SFW;backgroundOutputSettings" >
<!ENTITY % sliceNamingOutputSettingsElement "%NSPREFIX-SFW;sliceNamingOutputSettings" >
<!ENTITY % fileNamingOutputSettingsElement "%NSPREFIX-SFW;fileNamingOutputSettings" >
<!ENTITY % sliceNameElement "%NSPREFIX-SFW;sliceName" >
<!ENTITY % rolloverStateElement "%NSPREFIX-SFW;rolloverState" >
<!ENTITY % rolloverAbbrElement "%NSPREFIX-SFW;rolloverAbbr" >
<!ENTITY % triggerNameElement "%NSPREFIX-SFW;triggerName" >
<!ENTITY % triggerNoElement "%NSPREFIX-SFW;triggerNo" >
<!ENTITY % docNameElement "%NSPREFIX-SFW;docName" >
<!ENTITY % literal_sliceElement "%NSPREFIX-SFW;literal_slice" >
<!ENTITY % sliceNo_12Element "%NSPREFIX-SFW;sliceNo_12" >
<!ENTITY % sliceNo_0102Element "%NSPREFIX-SFW;sliceNo_0102" >
<!ENTITY % sliceNo_abElement "%NSPREFIX-SFW;sliceNo_ab" >
<!ENTITY % sliceNo_ABElement "%NSPREFIX-SFW;sliceNo_AB" >
<!ENTITY % date_mmddyyElement "%NSPREFIX-SFW;date_mmddyy" >
<!ENTITY % date_mmddElement "%NSPREFIX-SFW;date_mmdd" >
<!ENTITY % date_ddmmyyElement "%NSPREFIX-SFW;date_ddmmyy" >
<!ENTITY % date_ddmmElement "%NSPREFIX-SFW;date_ddmm" >
<!ENTITY % date_yyyymmddElement "%NSPREFIX-SFW;date_yyyymmdd" >
<!ENTITY % date_yymmddElement "%NSPREFIX-SFW;date_yymmdd" >
<!ENTITY % underscoreElement "%NSPREFIX-SFW;underscore" >
<!ENTITY % hyphenElement "%NSPREFIX-SFW;hyphen" >
<!ENTITY % spaceElement "%NSPREFIX-SFW;space" >
<!ENTITY % optimizationSettingsElement "%NSPREFIX-SFW;optimizationSettings" >
<!ENTITY % targetSettingsElement "%NSPREFIX-SFW;targetSettings" >
<!ENTITY % GIFFormatElement "%NSPREFIX-SFW;GIFFormat" >
<!ENTITY % PNG8FormatElement "%NSPREFIX-SFW;PNG8Format" >
<!ENTITY % PNG24FormatElement "%NSPREFIX-SFW;PNG24Format" >
<!ENTITY % JPEGFormatElement "%NSPREFIX-SFW;JPEGFormat" >
<!ENTITY % colorTableElement "%NSPREFIX-SFW;colorTable" >
<!ENTITY % colorElement "%NSPREFIX-SFW;color" >
<!ENTITY % lockedColorsElement "%NSPREFIX-SFW;lockedColors" >
<!ENTITY % colorShiftEntriesElement "%NSPREFIX-SFW;colorShiftEntries" >
<!ENTITY % sliceSourceBoundsElement "%NSPREFIX-SFW;sliceSourceBounds" >
<!ENTITY % webOptimizationDefsElement "%NSPREFIX-SFW;webOptimizationDefs" >
<!ENTITY % webOptimizationElement "%NSPREFIX-SFW;webOptimization" >
<!ENTITY % HTMLOptimizationElement "%NSPREFIX-SFW;HTMLOptimization" >
<!ENTITY % SVGOptimizationElement "%NSPREFIX-SFW;SVGOptimization" >
```

```
<!ENTITY % SWFOptimizationElement "%NSPREFIX-SFW;SWFOptimization" >
<!ENTITY % GIFOptimizationElement "%NSPREFIX-SFW;GIFOptimization" >
<!ENTITY % JPEGOptimizationElement "%NSPREFIX-SFW;JPEGOptimization" >
<!ENTITY % PNG8OptimizationElement "%NSPREFIX-SFW;PNG8Optimization" >
<!ENTITY % PNG24OptimizationElement "%NSPREFIX-SFW;PNG24Optimization" >

<!-- Data type declarations -->

<!ENTITY % background-value "CDATA">
   <!-- 'none', 'matte' or <CSS color> -->
<!ENTITY % boolean "CDATA">
<!ENTITY % color "CDATA">
<!ENTITY % color-list "CDATA">
   <!-- comma-separated list of CSS colors -->
<!ENTITY % horzAlign-value "CDATA">
   <!-- Numbers corresponding to (default|left|center|right) -->
<!ENTITY % integer "CDATA">
<!ENTITY % number "CDATA">
<!ENTITY % string "CDATA">
<!ENTITY % target-value "CDATA">
   <!-- (_blank|_self|_parent|_top) or empty string -->
<!ENTITY % vertAlign-value "CDATA">
   <!-- Numbers corresponding to (default|top|baseline|middle|bottom) -->
<!ENTITY % xpath "CDATA">


<!ELEMENT %sfwElement; (%slicesElement;|
               %resamplingParamsElement;|
               %outputSettingsElement;|
               %optimizationSettingsElement;|
               %sliceSourceBoundsElement;|
               %webOptimizationDefsElement;)* >
<!ATTLIST %sfwElement;
  %XMLNS-DECLARE-SFW;
  >

<!ELEMENT %slicesElement; (%sliceElement;)* >
<!ATTLIST %slicesElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %sliceElement; (%cellTextElement;)* >
<!ATTLIST %sliceElement;
  %XMLNS-DECLARE-SFW;
  webOptimization IDREF #IMPLIED
  colorTable %color-list; #IMPLIED
  sliceGroup CDATA #IMPLIED
  sliceType (user|object|auto) #IMPLIED
  associatedObject %xpath; #IMPLIED
  sliceName %string; #IMPLIED
  name %string; #IMPLIED
  type %string; #IMPLIED
  groupID %string; #IMPLIED
```

```
  sliceID CDATA #IMPLIED
  x %integer; #IMPLIED
  y %integer; #IMPLIED
  width %integer; #IMPLIED
  height %integer; #IMPLIED
  url CDATA #IMPLIED
  target %target-value; #IMPLIED
  message %string; #IMPLIED
  altTag %string; #IMPLIED
  cellTextIsHTML %boolean; #IMPLIED
  horzAlign %horzAlign-value; #IMPLIED
  vertAlign %vertAlign-value; #IMPLIED
  background %background-value; #IMPLIED
  >

<!ELEMENT %cellTextElement; (#PCDATA) >
<!ATTLIST %cellTextElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %resamplingParamsElement; ANY >
<!ATTLIST %resamplingParamsElement;
  %XMLNS-DECLARE-SFW;
  width %number; #IMPLIED
  height %number; #IMPLIED
  method (smooth|jagged) #IMPLIED >

<!ELEMENT %optimizationSettingsElement; (%targetSettingsElement;)+ >
<!ATTLIST %optimizationSettingsElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %outputSettingsElement; (%htmlOutputSettingsElement;|
                                   %backgroundOutputSettingsElement;|
                                   %sliceNamingOutputSettingsElement;|
                                   %fileNamingOutputSettingsElement;)* >
<!ATTLIST %outputSettingsElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %htmlOutputSettingsElement; ANY >
<!ATTLIST %htmlOutputSettingsElement;
  %XMLNS-DECLARE-SFW;
  tagsCase (AB|Ab|ab) #IMPLIED
  attrCase (AB|Ab|ab) #IMPLIED
  indent (none|tab|1|2|4|5|8) #IMPLIED
  newline (mac|unix|win) #IMPLIED
  quoteAttr %boolean; #IMPLIED
  includeComments %boolean; #IMPLIED
  generate (CSS|table) #IMPLIED
  cssReferencing (id|class|inline) #IMPLIED
  emptyCells (gif-image|gif-td|nowrap-td) #IMPLIED
  tdWH (auto|always|never) #IMPLIED
  spacerCells (auto|always|never) #IMPLIED >
```

```
<!ELEMENT %backgroundOutputSettingsElement; ANY >
<!ATTLIST %backgroundOutputSettingsElement;
  %XMLNS-DECLARE-SFW;
  viewAs (image|background) #IMPLIED
  image %string; #IMPLIED
  background %background-value; #IMPLIED >

<!ELEMENT %sliceNamingOutputSettingsElement; (%docNameElement;|
                    %literal_sliceElement;|
                    %sliceNo_12Element;|%sliceNo_0102Element;|
                    %sliceNo_abElement;|%sliceNo_ABElement;|
                    %date_mmddyyElement;|%date_mmddElement;|
                    %date_ddmmyyElement;|%date_ddmmElement;|
                    %date_yyyymmddElement;|%date_yymmddElement;|
                    %underscoreElement;|%hyphenElement;|
                    %spaceElement;)* >
<!ATTLIST %sliceNamingOutputSettingsElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %fileNamingOutputSettingsElement; (%sliceNameElement;|
                    %rolloverStateElement;|%rolloverAbbrElement;|
                    %triggerNameElement;|%triggerNoElement;|
                    %docNameElement;|
                    %sliceNo_12Element;|%sliceNo_0102Element;|
                    %sliceNo_abElement;|%sliceNo_ABElement;|
                    %date_mmddyyElement;|%date_mmddElement;|
                    %date_ddmmyyElement;|%date_ddmmElement;|
                    %date_yyyymmddElement;|%date_yymmddElement;|
                    %underscoreElement;|%hyphenElement;|
                    %spaceElement;)* >
<!ATTLIST %fileNamingOutputSettingsElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %docNameElement; ANY >
<!ATTLIST %docNameElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %literal_sliceElement; ANY >
<!ATTLIST %literal_sliceElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %sliceNo_12Element; ANY >
<!ATTLIST %sliceNo_12Element;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %sliceNo_0102Element; ANY >
<!ATTLIST %sliceNo_0102Element;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %sliceNo_abElement; ANY >
<!ATTLIST %sliceNo_abElement;
  %XMLNS-DECLARE-SFW; >
```

```
<!ELEMENT %sliceNo_ABElement; ANY >
<!ATTLIST %sliceNo_ABElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_mmddyyElement; ANY >
<!ATTLIST %date_mmddyyElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_mmddElement; ANY >
<!ATTLIST %date_mmddElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_ddmmyyElement; ANY >
<!ATTLIST %date_ddmmyyElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_ddmmElement; ANY >
<!ATTLIST %date_ddmmElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_yyyymmddElement; ANY >
<!ATTLIST %date_yyyymmddElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %date_yymmddElement; ANY >
<!ATTLIST %date_yymmddElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %underscoreElement; ANY >
<!ATTLIST %underscoreElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %hyphenElement; ANY >
<!ATTLIST %hyphenElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %spaceElement; ANY >
<!ATTLIST %spaceElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %targetSettingsElement; (%GIFFormatElement; |
                    %PNG8FormatElement;|
                    %PNG24FormatElement;|
                    %JPEGFormatElement;) >
<!ATTLIST %targetSettingsElement;
  %XMLNS-DECLARE-SFW;
  fileFormat CDATA #IMPLIED
  targetSettingsID CDATA #IMPLIED
  >

<!ELEMENT %GIFFormatElement; (%colorTableElement;|
```

```
                %lockedColorsElement;|%colorShiftEntriesElement;)* >
<!ATTLIST %GIFFormatElement;
  %XMLNS-DECLARE-SFW;
  transparency CDATA #IMPLIED
  includeCaption CDATA #IMPLIED
  interlaced CDATA #IMPLIED
  noMatteColor CDATA #IMPLIED
  matteColor CDATA #IMPLIED
  autoReduce CDATA #IMPLIED
  rolloverMasterPalette CDATA #IMPLIED
  webShiftPercent CDATA #IMPLIED
  numColors CDATA #IMPLIED
  lossy CDATA #IMPLIED
  ditherAlgorithm CDATA #IMPLIED
  ditherPercent CDATA #IMPLIED
  reductionAlgorithm CDATA #IMPLIED
  >

<!ELEMENT %colorTableElement; (%colorElement;)* >
<!ATTLIST %colorTableElement;
  %XMLNS-DECLARE-SFW;
  isExact %boolean; #IMPLIED
  >

<!ELEMENT %lockedColorsElement; (%colorElement;)* >
<!ATTLIST %lockedColorsElement;
  %XMLNS-DECLARE-SFW;
  >

<!ELEMENT %colorShiftEntriesElement; (%colorElement;)* >
<!ATTLIST %colorShiftEntriesElement;
  %XMLNS-DECLARE-SFW;
  >

<!ELEMENT %colorElement; ANY >
<!ATTLIST %colorElement;
  %XMLNS-DECLARE-SFW;
  rgb %color; #IMPLIED
  rangeStart %color; #IMPLIED
  rangeEnd %color; #IMPLIED
  remapColor %color; #IMPLIED
  >

<!ELEMENT %PNG8FormatElement; ANY >
<!ATTLIST %PNG8FormatElement;
  %XMLNS-DECLARE-SFW;
  transparency CDATA #IMPLIED
  includeCaption CDATA #IMPLIED
  interlaced CDATA #IMPLIED
  noMatteColor CDATA #IMPLIED
  matteColor CDATA #IMPLIED
  filtered CDATA #IMPLIED
```

```
  rolloverMasterPalette CDATA #IMPLIED
  webShiftPercent CDATA #IMPLIED
  numColors CDATA #IMPLIED
  lossy CDATA #IMPLIED
  ditherAlgorithm CDATA #IMPLIED
  ditherPercent CDATA #IMPLIED
  reductionAlgorithm CDATA #IMPLIED
  >

<!ELEMENT %PNG24FormatElement; ANY >
<!ATTLIST %PNG24FormatElement;
  %XMLNS-DECLARE-SFW;
  transparency CDATA #IMPLIED
  interlaced CDATA #IMPLIED
  noMatteColor CDATA #IMPLIED
  matteColor CDATA #IMPLIED
  filtered CDATA #IMPLIED
  >

<!ELEMENT %JPEGFormatElement; ANY >
<!ATTLIST %JPEGFormatElement;
  %XMLNS-DECLARE-SFW;
  noMatteColor CDATA #IMPLIED
  matteColor CDATA #IMPLIED
  optimized CDATA #IMPLIED
  quality CDATA #IMPLIED
  progressive CDATA #IMPLIED
  numColors CDATA #IMPLIED
  blurAmount CDATA #IMPLIED
  embedICCProfile CDATA #IMPLIED
  >

<!ELEMENT sliceSourceBounds EMPTY >
<!ATTLIST sliceSourceBounds
  %XMLNS-DECLARE-SFW;
  x CDATA #IMPLIED
  y CDATA #IMPLIED
  width CDATA #IMPLIED
  height CDATA #IMPLIED
  bottomLeftOrigin CDATA #IMPLIED
  >

<!ELEMENT %webOptimizationDefsElement; (%webOptimizationElement;)* >
<!ATTLIST %webOptimizationDefsElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %webOptimizationElement; (%HTMLOptimizationElement;|
                    %SVGOptimizationElement;|%SWFOptimizationElement;|
                    %GIFOptimizationElement;|%JPEGOptimizationElement;|
                    %PNG8OptimizationElement;|%PNG24OptimizationElement;) >
<!ATTLIST %webOptimizationElement;
  %XMLNS-DECLARE-SFW;
```

```
  id ID #IMPLIED >

<!ELEMENT %HTMLOptimizationElement; ANY >
<!ATTLIST %HTMLOptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %SVGOptimizationElement; ANY >
<!ATTLIST %SVGOptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %SWFOptimizationElement; ANY >
<!ATTLIST %SWFOptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %GIFOptimizationElement; ANY >
<!ATTLIST %GIFOptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %JPEGOptimizationElement; ANY >
<!ATTLIST %JPEGOptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %PNG8OptimizationElement; ANY >
<!ATTLIST %PNG8OptimizationElement;
  %XMLNS-DECLARE-SFW; >

<!ELEMENT %PNG24OptimizationElement; ANY >
<!ATTLIST %PNG24OptimizationElement;
  %XMLNS-DECLARE-SFW; >
```

4.5 Example

The following is sample XML for the SFW XML grammar embedded in an SVG file exported by Adobe Illustrator 10.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 10.0...  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
      "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd" [
  ...
  <!ENTITY ns_sfw "http://ns.adobe.com/SaveForWeb/1.0/">
  <!ENTITY ns_svg "http://www.w3.org/2000/svg">
  <!ENTITY ns_xlink "http://www.w3.org/1999/xlink">
  ...
]>
<svg xmlns="&ns_svg;" xmlns:xlink="&ns_xlink;"
   width="436.358" height="692.515" viewBox="0 0 436.358 692.515"
   ... >
  <metadata>
    <sfw  xmlns="&ns_sfw;">
      <slices>
```

```
        <slice y="585" x="115" name="" type="1"
               width="103" height="84" groupID="419474360"
               url="" target="" message="" altTag=""
               cellTextIsHTML="false" horzAlign="1"
               vertAlign="1" background="none" sliceID="16807">
          <cellText>
            <![CDATA[]]>
          </cellText>
        </slice>
        ...
      </slices>
      <sliceSourceBounds  y="6.074" x="8.028"
                          width="436.358" height="692.515"
                          bottomLeftOrigin="true">
      </sliceSourceBounds>
      <optimizationSettings>
        <targetSettings  fileFormat="GIFFormat" targetSettingsID="0">
            <GIFFormat  transparency="true" includeCaption="false"
                        interlaced="false" noMatteColor="false"
                        matteColor="#FFFFFF" autoReduce="false"
                        rolloverMasterPalette="false"
                        webShiftPercent="0" numColors="256"
                        lossy="0" ditherAlgorithm="diffusion"
                        ditherPercent="100"
                        reductionAlgorithm="selective">
            <lockedColors>
              <color  rgb="#FF105D"></color>
              <color  rgb="#FF3073"></color>
            </lockedColors>
            <colorShiftEntries>
              <color  rangeStart="#FF105D" rangeEnd="#FF105D"
                      remapColor="#FF12FF"></color>
              <color  rangeStart="#FF3073" rangeEnd="#FF3073"
                      remapColor="#160209"></color>
            </colorShiftEntries>
          </GIFFormat>
        </targetSettings>
        ...
      </optimizationSettings>
    </sfw>
  </metadata>
  <g id="Layer_1">
    <path fill="#FF0052" d="..."/>
    ...
    <path id="_x003C_Slice_x003E_" fill="none" d="..."/>
  </g>
</svg>
```

5 Image Replacement XML Grammar

5.1 About this Chapter

This chapter describes the XML grammar for replaceable images.

5.2 About Replaceable Images

Users often need the ability to place an image in an EPS or SVG file and have that image be replaced with another image stored in a separate file. Adobe Illustrator 10 and AlterCast™ 1.5 support the ability to do that and allow it to be data-driven. With Illustrator, you can assign an "image replace" variable to an image object, and then in AlterCast you can change the variable to point to a server-side image, thereby doing server-side, data-driven image replacement. Ilustrator will offer various UI options for controlling image replacement parameters, such as whether the replacement image should shrink to fit into the same area. These parameters and other supplemental information are expressed in the grammar shown below.

5.2.1 Image Replacement Definitions

The following are a few basic definitions for image replacement.

reference rectangle	The rectangular area into which the replacement image is to be placed.
replacement image	The image in an external file that is to replace the image in the document.
original image	The image in the document, which is to be replaced by the replacement image.

5.3 Namespace

The following is the namespace URI for image replacement:

```
http://ns.adobe.com/ImageReplacement/1.0/
```

5.4 DTD

The following is the DTD for Image Replacement:

```
<!-- ImageReplacement.dtd (version 1.0) 20010915 -->

<!-- This DTD summarizes the Image Replacement namespace
     exported from Adobe Illustrator 10.0 and supported by
     AlterCast 1.0.
     The namespace URI is "http://ns.adobe.com/ImageReplacement/1.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-IR (Namespace prefix for this namespace, such as "imrep:")
       XMLNS-DECLARE-IR (xmlns[:prefix] attribute declaration)
     and then include this DTD via reference. -->


<!-- Data types -->
<!ENTITY % boolean "(true|false)" >
<!ENTITY % integer-positive "CDATA" >
<!ENTITY % number "CDATA" >
<!ENTITY % number-nonnegative "CDATA" >
<!ENTITY % URI "CDATA" >

<!ENTITY % imageReplacementElement "%NSPREFIX-IR;imageReplacement" >

<!ELEMENT %imageReplacementElement; ANY >
<!ATTLIST %imageReplacementElement;
  %XMLNS-DECLARE-IR;
  id ID #IMPLIED
  x %number; #IMPLIED
  y %number; #IMPLIED
  width %number-nonnegative; #REQUIRED
  height %number-nonnegative; #REQUIRED
  refWidth %integer-positive; #REQUIRED
  refHeight %integer-positive; #REQUIRED
  align (left|center|right) 'left'
  valign (top|middle|bottom) 'top'
  placementMethod (conform|fit|fill|ratio) 'conform' >
```

Attribute definitions:

`id = "<name>"`

Standard ‘id’ attribute found in most XML grammars.

x = "%number;"

The x-axis coordinate of the side of the reference rectangle which has the smaller x-axis coordinate value in the current user coordinate system. If the attribute is not specified, the effect is as if a value of “0” were specified.

y = "%number;"

Corresponding y-axis coordinate for the reference rectangle.

width = "%integer-positive;"

The width of the reference rectangle expressed as a length value in the current user coordinate system. If not specified or if a negative value is supplied, then on any attempt to perform an image replacement an error message will be generated and the result of the image replacement operation will be as if no image were rendered. A zero value will not cause an error message, and the result of the image replacement operation will be as if no image were rendered.

height = "%number-nonnegative;"

Corresponding height of the reference rectangle.

refWidth = "%integer-positive;"

The width in pixels of the original image (i.e., the image which will be replaced). This `refWidth` will be used in some cases to perform ratio calculations in support of some of the ‘`placementMethod`’ options. If not specified, or if an out-of-range value is supplied, then on any attempt to perform an image replacement, an error message will be generated and the result of the image replacement operation will be as if no image were rendered.

refHeight = "%integer-positive;"

The corresponding height in pixels of the original image.

align = "(left|center|right)"

The horizontal alignment for both the reference rectangle and the replacement image. For example, a value of **`left`** indicates that the left edge of the replacement image will be aligned with the left edge of the reference rectangle. Note that because of SVG’s default coordinate system having its positive x-axis pointing right, a value of **`left`** in SVG’s default coordinate system will indeed represent the leftmost edge visually. However, to take into account the possibility of coordinate system transformations, the formal definition is that **`left`** corresponds to the minimum x-axis edge in user space and that **`right`** corresponds to the maximum x-axis edge in user space. If the attribute is not specified, the effect is as if a value of **`left`** were specified.

`valign = "(top|middle|bottom)"`

The vertical alignment for both the reference rectangle and the replacement image. For example, a value of **`top`** indicates that the top edge of the replacement image will be aligned with the top edge of the reference rectangle. Note that because of SVG's default coordinate system having its positive y-axis pointing down, a value of **`top`** in SVG's default coordinate system will indeed represent the topmost edge visually. However, to take into account the possibility of coordinate system transformations, the formal definition is that **`top`** corresponds to the minimum y-axis edge in userspace and that **`bottom`** corresponds to the maximum y-axis edge in userspace. If the attribute is not specified, the effect is as if a value of *top* were specified.

`placementMethod = "(conform|fit|fill|ratio)"`

This attribute determines how the replacement image is scaled to fit relative to the reference rectangle and possible relative to the ratios of the pixel resolutions of the original image (i.e., the image which will be replaced) and the replacement image. See below for an illustration of the effect which should be produced by the four options. Possible values are:

`conform`	The replacement image is stretched (potentially non-uniformly) to fit into the reference rectangle exactly.
`fit`	The replacement image is stretched uniformly (i.e., same supplementary scale factor is applied to both axes) to fit into the reference rectangle exactly such that one dimension spans the entire reference rectangle and the other dimension is no larger than the reference rectangle. Therefore, it is possible that parts of the reference rectangle will be uncovered by the replacement image.
`fill`	The replacement image is stretched uniformly (i.e., same supplementary scale factor is applied to both axes) to fit into the reference rectangle exactly such that one dimension spans the entire reference rectangle and the other dimension is no smaller than the reference rectangle. Therefore, it is possible that the replacement image might extend outside of reference rectangle in one axis.
`ratio`	The replacement image will use the same scale factor as the original image. Thus, if the replacement image has a pixel width of *pw* and a pixel height of *ph*, then the replacement image's width on the canvas will be **`pw`**/**`refWidth`** as large as the the original image's width on the canvas, and the replacement image's height on the canvas will be **`ph`**/**`refHeight`** as large as the the original image's height on the canvas.

5.5 Example

The following source code shows an example of how to define a replaceable image object within an SVG file.

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
  "http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg-20001102.dtd" [
<!ENTITY ns_imrep "http://ns.adobe.com/ImageReplacement/1.0/">
<!ENTITY ns_adobe_gen "http://ns.adobe.com/AdobeGeneral/1.0/">
]>
<svg  width="336.333pt" height="328.346pt" viewBox="0 0 336.333 328.346">
  <switch i:objectNS="&ns_imrep;" i:objectType="replaceable-image">
    <foreignObject x="100" y="100" width="200" height="150"
                  requiredExtensions="&ns_imrep;" overflow="visible">
      <imageReplacement xmlns="&ns_imrep;"  x="100" y="100" width="200" height="150"
                        align="left" valign="top" refWidth="300" refHeight="600"
                        placementMethod="conform"/>
      <a:targetRef xlink:href="#c350438"/>
    </foreignObject>
    <image id="c350438" x="100" y="100" width="200" height="150"
          overflow="visible" xlink:href="foo.jpg"/>
  </switch>
</svg>
```

6 Graphing XML Grammar

6.1 About This Chapter

This chapter describes the XML grammar for graphs exported in SVG files by Adobe Illustrator 10.

6.2 About Graphs

The graphing module matches the feature set in the graphing module in AlterCast 1.5. Illustrator 10 will export its current graphing features using this grammar when particular SVG export options are set (e.g., to include AlterCast information).

The Graphing XML grammar is defined in “Adobe AlterCast Commands Reference,” which will be available when AlterCast 1.5 is published.

6.3 Namespace

The following is the namespace URI for graphs:

```
http://ns.adobe.com/Graphs/1.0/
```

6.4 DTD

The following is the DTD for the graphing namespace:

```
<!-- Graphs.dtd (version 1.0) 20010915 -->

<!-- This DTD summarizes the Graphs namespace
     exported from Adobe Illustrator 10.0 and supported by
     AlterCast 1.0.
     The namespace URI is "http://ns.adobe.com/Graphs/1.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-GRAPHS (Namespace prefix for this namespace, such as "g:")
       XMLNS-DECLARE-GRAPHS (xmlns[:prefix] attribute declaration)
         and then include this DTD via reference. -->

<!-- Data type declarations -->

<!ENTITY % integer "CDATA">
<!ENTITY % number "CDATA">
```

```
<!ENTITY % number-list "CDATA">

<!-- orientation is an enum of (+|-), but one of the choices, "+",
     is illegal in DTDs, so it defined as CDATA here. -->
<!ENTITY % orientationValue "CDATA" >

<!-- positionBase and positionOffset can be either a number
     or an enum of (minEdge|maxEdge). -->
<!ENTITY % positionValue "CDATA" >

<!-- tickLength can be either a number
     or an enum of (none|short|long). -->
<!ENTITY % tickLengthValue "CDATA" >

<!-- graphDescTypeValue is any string, but first
     version only supports (bar|point|pie). -->
<!ENTITY % graphDescTypeValue "CDATA" >


<!ENTITY % applyToAllElement "%NSPREFIX-GRAPHS;applyToAll" >
<!ENTITY % applyToAllInGraphElement "%NSPREFIX-GRAPHS;applyToAllInGraph" >
<!ENTITY % applyToLegendElement "%NSPREFIX-GRAPHS;applyToLegend" >
<!ENTITY % axisElement "%NSPREFIX-GRAPHS;axis" >
<!ENTITY % axisNamesElement "%NSPREFIX-GRAPHS;axisNames" >
<!ENTITY % axisStringElement "%NSPREFIX-GRAPHS;axisString" >
<!ENTITY % calculatedElement "%NSPREFIX-GRAPHS;calculated" >
<!ENTITY % changeApplyDesignElement "%NSPREFIX-GRAPHS;changeApplyDesign" >
<!ENTITY % changeDeleteElement "%NSPREFIX-GRAPHS;changeDelete" >
<!ENTITY % changeGraphicStyleElement "%NSPREFIX-GRAPHS;changeGraphicStyle" >
<!ENTITY % changeHideElement "%NSPREFIX-GRAPHS;changeHide" >
<!ENTITY % changeStackingElement "%NSPREFIX-GRAPHS;changeStacking" >
<!ENTITY % changeTagElement "%NSPREFIX-GRAPHS;changeTag" >
<!ENTITY % changeTextStyleElement "%NSPREFIX-GRAPHS;changeTextStyle" >
<!ENTITY % changeTransformElement "%NSPREFIX-GRAPHS;changeTransform" >
<!ENTITY % dataColumnElement "%NSPREFIX-GRAPHS;dataColumn" >
<!ENTITY % dataColumnsElement "%NSPREFIX-GRAPHS;dataColumns" >
<!ENTITY % dataElement "%NSPREFIX-GRAPHS;data" >
<!ENTITY % dictElement "%NSPREFIX-GRAPHS;dict" >
<!ENTITY % differenceElement "%NSPREFIX-GRAPHS;difference" >
<!ENTITY % entryElement "%NSPREFIX-GRAPHS;entry" >
<!ENTITY % formatElement "%NSPREFIX-GRAPHS;format" >
<!ENTITY % globalMapElement "%NSPREFIX-GRAPHS;globalMap" >
<!ENTITY % graphElement "%NSPREFIX-GRAPHS;graph" >
<!ENTITY % graphDescElement "%NSPREFIX-GRAPHS;graphDesc" >
<!ENTITY % ignoredDataColumnsElement "%NSPREFIX-GRAPHS;ignoredDataColumns" >
<!ENTITY % intElement "%NSPREFIX-GRAPHS;int" >
<!ENTITY % intersectionElement "%NSPREFIX-GRAPHS;intersection" >
<!ENTITY % labelStringElement "%NSPREFIX-GRAPHS;labelString" >
<!ENTITY % legendElement "%NSPREFIX-GRAPHS;legend" >
<!ENTITY % legendStringElement "%NSPREFIX-GRAPHS;legendString" >
<!ENTITY % majorAxisElement "%NSPREFIX-GRAPHS;majorAxis" >
<!ENTITY % majorDataColumnsElement "%NSPREFIX-GRAPHS;majorDataColumns" >
```

```
<!ENTITY % mapElement "%NSPREFIX-GRAPHS;map" >
<!ENTITY % minorAxisElement "%NSPREFIX-GRAPHS;minorAxis" >
<!ENTITY % minorDataColumnsElement "%NSPREFIX-GRAPHS;minorDataColumns" >
<!ENTITY % nameElement "%NSPREFIX-GRAPHS;name" >
<!ENTITY % nthElement "%NSPREFIX-GRAPHS;nth" >
<!ENTITY % nthFromEndElement "%NSPREFIX-GRAPHS;nthFromEnd" >
<!ENTITY % nthFromStartElement "%NSPREFIX-GRAPHS;nthFromStart" >
<!ENTITY % nthRepeatElement "%NSPREFIX-GRAPHS;nthRepeat" >
<!ENTITY % nthRepeatFromEndElement "%NSPREFIX-GRAPHS;nthRepeatFromEnd" >
<!ENTITY % nthRepeatFromStartElement "%NSPREFIX-GRAPHS;nthRepeatFromStart" >
<!ENTITY % otherElement "%NSPREFIX-GRAPHS;other" >
<!ENTITY % propertyElement "%NSPREFIX-GRAPHS;property" >
<!ENTITY % propertyRowElement "%NSPREFIX-GRAPHS;propertyRow" >
<!ENTITY % rowElement "%NSPREFIX-GRAPHS;row" >
<!ENTITY % simplePropsElement "%NSPREFIX-GRAPHS;simpleProps" >
<!ENTITY % transformFillElement "%NSPREFIX-GRAPHS;transformFill" >
<!ENTITY % transformFilterElement "%NSPREFIX-GRAPHS;transformFilter" >
<!ENTITY % transformObjectElement "%NSPREFIX-GRAPHS;transformObject" >
<!ENTITY % transformStrokeElement "%NSPREFIX-GRAPHS;transformStroke" >
<!ENTITY % transformStrokeWidthElement "%NSPREFIX-GRAPHS;transformStrokeWidth" >
<!ENTITY % unionElement "%NSPREFIX-GRAPHS;union" >
<!ENTITY % valueElement "%NSPREFIX-GRAPHS;value" >
<!ENTITY % valuesElement "%NSPREFIX-GRAPHS;values" >
<!ENTITY % whatElement "%NSPREFIX-GRAPHS;what" >
<!ENTITY % whatToTransformElement "%NSPREFIX-GRAPHS;whatToTransform" >
<!ENTITY % whichElement "%NSPREFIX-GRAPHS;which" >
<!ENTITY % whichAxisElement "%NSPREFIX-GRAPHS;whichAxis" >
<!ENTITY % whichAxisIsSubpartElement "%NSPREFIX-GRAPHS;whichAxisIsSubpart" >
<!ENTITY % whichPointElement "%NSPREFIX-GRAPHS;whichPoint" >
<!ENTITY % whichSeriesElement "%NSPREFIX-GRAPHS;whichSeries" >
<!ENTITY % xformElement "%NSPREFIX-GRAPHS;xform" >
<!ENTITY % xformsElement "%NSPREFIX-GRAPHS;xforms" >


<!ELEMENT %graphElement; ((%dataElement;),(%formatElement;)) >
<!ATTLIST %graphElement;
  %XMLNS-DECLARE-GRAPHS;
  >

<!ELEMENT %dataElement; (%propertyRowElement;|%mapElement;|%globalMapElement;|
                  %propertyElement;|%valuesElement;|%dataColumnElement;)* >
<!ATTLIST %dataElement;
  %XMLNS-DECLARE-GRAPHS;
  numDataColumns %integer; #REQUIRED
  >

<!ELEMENT %propertyRowElement; (%valueElement;)* >
<!ATTLIST %propertyRowElement;
  key CDATA #IMPLIED
  >

<!ELEMENT %propertyElement; (%valueElement;)* >
```

```
<!ATTLIST %propertyElement;
  key CDATA #IMPLIED
  >

<!ELEMENT %mapElement; (%entryElement;)* >
<!ATTLIST %mapElement;
  column %integer; #IMPLIED
  key CDATA #IMPLIED
  >

<!ELEMENT %globalMapElement; (%entryElement;)* >
<!ATTLIST %globalMapElement;
  key CDATA #IMPLIED
  >

<!ELEMENT %valuesElement; (%rowElement;)* >
<!ATTLIST %valuesElement; >

<!ELEMENT %rowElement; (%valueElement;)* >
<!ATTLIST %rowElement; >

<!ELEMENT %valueElement; (#PCDATA) >
<!ATTLIST %valueElement;
  key CDATA #IMPLIED
  type CDATA #IMPLIED
  >

<!ELEMENT %dataColumnElement; (%simplePropsElement;) >
<!ATTLIST %dataColumnElement;
  columnNumber CDATA #REQUIRED
  >

<!ELEMENT %simplePropsElement; (%dictElement;) >
<!ATTLIST %simplePropsElement; >

<!ELEMENT %dictElement; (%entryElement;)* >
<!ATTLIST %dictElement; >

<!ELEMENT %entryElement; (#PCDATA) >
<!ATTLIST %entryElement;
  key CDATA #REQUIRED
  >

<!-- invert %integer; #IMPLIED not supported in version 1 -->
<!ELEMENT %formatElement; (%majorAxisElement;|%minorAxisElement;|%axisElement;
                           |%graphDescElement;|%legendElement;|%xformsElement;
                           |%ignoredDataColumnsElement;)* >
<!ATTLIST %formatElement;
  %XMLNS-DECLARE-GRAPHS;
  type (2D) #REQUIRED
  width %number; #REQUIRED
  height %number; #REQUIRED
```

```
  axisLayout (cartesian|polar) #REQUIRED
  originDegrees %integer; #IMPLIED
  ignoredDataColumns %number-list; #IMPLIED
      seriesStackingOrder (firstInFront|lastInFront) #IMPLIED
  inSeriesStackingOrder (firstInFront|lastInFront) #IMPLIED
  >

<!ELEMENT %majorAxisElement; (%dataColumnsElement;)? >
<!ATTLIST %majorAxisElement;
  type (enumeration|value) #IMPLIED
      direction (x|y|polar|radius|circumpherence) #REQUIRED
  orientation %orientationValue; #IMPLIED
  valueConflict (sum|product|max|min|count|separate) #IMPLIED
  >

<!ELEMENT %minorAxisElement; (%dataColumnsElement;)? >
<!ATTLIST %minorAxisElement;
  type (enumeration|value) #IMPLIED
      direction (x|y|polar|radius|circumpherence) #REQUIRED
  orientation %orientationValue; #IMPLIED
  >

<!ELEMENT %dataColumnsElement; (%intElement;|%otherElement;|%calculatedElement;)* >
<!ATTLIST %dataColumnsElement; >

<!ELEMENT %majorDataColumnsElement; (%intElement;|%otherElement;|%calculatedElement;)*
>
<!ATTLIST %majorDataColumnsElement; >

<!ELEMENT %minorDataColumnsElement; (%intElement;|%otherElement;|%calculatedElement;)*
>
<!ATTLIST %minorDataColumnsElement; >

<!ELEMENT %intElement; (#PCDATA) >
<!ATTLIST %intElement; >

<!ELEMENT %otherElement; (#PCDATA) >
<!ATTLIST %otherElement; >

<!ELEMENT %calculatedElement; ANY >
<!ATTLIST %calculatedElement;
  base CDATA #IMPLIED
  repeat CDATA #IMPLIED >

<!ELEMENT %ignoredDataColumnsElement; (%intElement;) >
<!ATTLIST %ignoredDataColumnsElement; >

<!-- axisString not supported in first version -->
<!-- following properties are not supported in first version
  minBase %number; #IMPLIED
  minOffset %number; #IMPLIED
  maxBase %number; #IMPLIED
```

```
  maxOffset %number; #IMPLIED
 -->
<!ELEMENT %axisElement; (%labelStringElement;|%axisStringElement;)* >
<!ATTLIST %axisElement;
  type (major|minor) #REQUIRED
  name CDATA #REQUIRED
  positionBase %positionValue; #IMPLIED
  positionOffset %positionValue; #IMPLIED
  repeat %number; #IMPLIED
  ticksAndLabels (explicit|aiDefaultValue) #REQUIRED
  axisVisibility (visible|none) #IMPLIED
  basis %integer; #IMPLIED
  tickLength %tickLengthValue; #IMPLIED
  tickPositionBase %number; #IMPLIED
  tickPositionOffset %number; #IMPLIED
  tickSpacing %number; #IMPLIED
  subTicks %number; #IMPLIED
  subTickLength %tickLengthValue; #IMPLIED
  labelPositionBase %number; #IMPLIED
  labelPositionOffset %number; #IMPLIED
  labelSpacing %number; #IMPLIED
  >

<!ELEMENT %labelStringElement; (#PCDATA) >
<!ATTLIST %labelStringElement; >

<!ELEMENT %axisStringElement; (#PCDATA) >
<!ATTLIST %axisStringElement; >

<!-- invert property not supported in version 1 -->
<!ELEMENT %graphDescElement; (%legendStringElement;|%majorDataColumnsElement;|
                    %minorDataColumnsElement;|%axisNamesElement;)* >
<!ATTLIST %graphDescElement;
  type %graphDescTypeValue; #IMPLIED
  majorRange (explicit|span|paddedSpan|aiDefaultValue|aiDefaultScatterValue) #IMPLIED
  majorMinBase %number; #IMPLIED
  majorMinOffset %number; #IMPLIED
  majorMaxBase %number; #IMPLIED
  majorMaxOffset %number; #IMPLIED
  minorRange (explicit|span|paddedSpan|aiDefaultValue|aiDefaultScatterValue) #IMPLIED
  minorMinBase %number; #IMPLIED
  minorMinOffset %number; #IMPLIED
  minorMaxBase %number; #IMPLIED
  minorMaxOffset %number; #IMPLIED
  dropShadow (none|simple) #IMPLIED
  order %number; #IMPLIED
  stacking (none|stacked) #IMPLIED
  normalizedStackTotal %number; #IMPLIED
  accumulation (none|sum|product|max|min) #IMPLIED
  sorting (none|firstAscending|firstDescending|eachAscending|eachDescending) #IMPLIED
  granularity %number; #IMPLIED
  columnWidth %number; #IMPLIED
```

```
  clusterWidth %number; #IMPLIED
  points (none|square|round) #IMPLIED
  pointSize %number; #IMPLIED
  fill (none|solid) #IMPLIED
  lineStyle (none|stroked|filled) #IMPLIED
  lineWidth %number; #IMPLIED
  multiLayout (proportional|equal|stacked) #IMPLIED
  >

<!ELEMENT %legendElement; ANY >
<!ATTLIST %legendElement;
      position (top|right|left|bottom|inside) #IMPLIED
  order (normal|reversed) #IMPLIED
  >

<!ELEMENT %legendStringElement; (#PCDATA) >
<!ATTLIST %legendStringElement; >

<!ELEMENT %axisNamesElement; (%nameElement;)* >
<!ATTLIST %axisNamesElement; >

<!ELEMENT %nameElement; (#PCDATA) >
<!ATTLIST %nameElement; >

<!ELEMENT %xformsElement; (%xformElement;)* >
<!ATTLIST %xformsElement; >

<!ELEMENT %xformElement;
(%whatElement;|%changeTransformElement;|%changeStackingElement;
                  |%changeDeleteElement;|%changeHideElement;|%changeTagElement;
                  |%changeGraphicStyleElement;|%changeTextStyleElement;
                  |%changeApplyDesignElement;)* >
<!ATTLIST %xformElement; >

<!ELEMENT %whatElement; (%whichSeriesElement;|%whichPointElement;|%whichAxisElement;
                 |%whichAxisIsSubpartElement;|%unionElement;|%intersectionElement;
                 |%differenceElement;)* >
<!ATTLIST %whatElement;
  xformBase (everything|legendText|graphics|fill|stroke|
             point|graphicsBase|axis|axisSpine|
             axisNormalTick|axisSubTick|axisLabel|
             axisString|shadowGraphics) #IMPLIED
  axisName CDATA #IMPLIED
  aggregate CDATA #IMPLIED
 >

<!ELEMENT %whichElement; (%whichSeriesElement;|%whichPointElement;)* >
<!ATTLIST %whichElement; >

<!ELEMENT %whichSeriesElement; (%applyToAllElement;|%applyToAllInGraphElement;
                |%applyToLegendElement;|%nthElement;|%nthFromEndElement;
                |%nthFromStartElement;|%nthRepeatElement;
```

```
                |%nthRepeatFromEndElement;|%nthRepeatFromStartElement;)* >
<!ATTLIST %whichSeriesElement;
 >

<!ELEMENT %whichPointElement; (%applyToAllElement;|%applyToAllInGraphElement;
                |%applyToLegendElement;|%nthElement;|%nthFromEndElement;
                |%nthFromStartElement;|%nthRepeatElement;
                |%nthRepeatFromEndElement;|%nthRepeatFromStartElement;)* >
<!ATTLIST %whichPointElement;
 >

<!ELEMENT %whichAxisElement; (%applyToAllElement;|%applyToAllInGraphElement;
                |%applyToLegendElement;|%nthElement;|%nthFromEndElement;
                |%nthFromStartElement;|%nthRepeatElement;
                |%nthRepeatFromEndElement;|%nthRepeatFromStartElement;)* >
<!ATTLIST %whichAxisElement;
 >

<!ELEMENT %whichAxisIsSubpartElement; (%applyToAllElement;|%applyToAllInGraphElement;
                |%applyToLegendElement;|%nthElement;|%nthFromEndElement;
                |%nthFromStartElement;|%nthRepeatElement;
                |%nthRepeatFromEndElement;|%nthRepeatFromStartElement;)* >
<!ATTLIST %whichAxisIsSubpartElement;
 >

<!ELEMENT %unionElement; (%whatElement;)* >
<!ATTLIST %unionElement; >

<!ELEMENT %intersectionElement; (%whatElement;)* >
<!ATTLIST %intersectionElement; >

<!ELEMENT %differenceElement; (%whatElement;)* >
<!ATTLIST %differenceElement; >

<!ELEMENT %applyToAllElement; ANY >
<!ATTLIST %applyToAllElement; >

<!ELEMENT %applyToAllInGraphElement; ANY >
<!ATTLIST %applyToAllInGraphElement; >

<!ELEMENT %applyToLegendElement; ANY >
<!ATTLIST %applyToLegendElement; >

<!ELEMENT %nthElement; ANY >
<!ATTLIST %nthElement; >

<!ELEMENT %nthFromStartElement; ANY >
<!ATTLIST %nthFromStartElement; >

<!ELEMENT %nthFromEndElement; ANY >
<!ATTLIST %nthFromEndElement; >
```

```
<!ELEMENT %nthRepeatElement; ANY >
<!ATTLIST %nthRepeatElement;
  start CDATA #IMPLIED
  repeat CDATA #IMPLIED >

<!ELEMENT %nthRepeatFromStartElement; ANY >
<!ATTLIST %nthRepeatFromStartElement;
  start CDATA #IMPLIED
  repeat CDATA #IMPLIED >

<!ELEMENT %nthRepeatFromEndElement; ANY >
<!ATTLIST %nthRepeatFromEndElement;
  start CDATA #IMPLIED
  repeat CDATA #IMPLIED >

<!ELEMENT %changeTransformElement; (%whatToTransformElement;)* >
<!ATTLIST %changeTransformElement;
  matrix CDATA #IMPLIED >

<!ELEMENT %whatToTransformElement; (%transformObjectElement;|%transformFillElement;

|%transformStrokeElement;|%transformStrokeWidthElement;|%transformFilterElement;)* >
<!ATTLIST %whatToTransformElement; >

<!ELEMENT %transformObjectElement; ANY >
<!ATTLIST %transformObjectElement; >

<!ELEMENT %transformFillElement; ANY >
<!ATTLIST %transformFillElement; >

<!ELEMENT %transformStrokeElement; ANY >
<!ATTLIST %transformStrokeElement; >

<!ELEMENT %transformStrokeWidthElement; ANY >
<!ATTLIST %transformStrokeWidthElement; >

<!ELEMENT %transformFilterElement; ANY >
<!ATTLIST %transformFilterElement; >

<!ELEMENT %changeStackingElement; ANY >
<!ATTLIST %changeStackingElement;
  stack CDATA #IMPLIED >

<!ELEMENT %changeDeleteElement; ANY >
<!ATTLIST %changeDeleteElement; >

<!ELEMENT %changeHideElement; ANY >
<!ATTLIST %changeHideElement;
  showHide CDATA #IMPLIED >

<!ELEMENT %changeTagElement; ANY >
<!ATTLIST %changeTagElement;
```

```
  value CDATA #IMPLIED >

<!ELEMENT %changeGraphicStyleElement; ANY >
<!ATTLIST %changeGraphicStyleElement;
  value CDATA #IMPLIED >

<!ELEMENT %changeTextStyleElement; ANY >
<!ATTLIST %changeTextStyleElement;
  value CDATA #IMPLIED >

<!ELEMENT %changeApplyDesignElement; ANY >
<!ATTLIST %changeApplyDesignElement;
  value CDATA #IMPLIED
  designApplication CDATA #IMPLIED
  legendOrientation CDATA #IMPLIED
  repeatValue CDATA #IMPLIED
  repeatRemainder CDATA #IMPLIED >
```

6.5 Example

The following is sample XML for the graphing XML grammar embedded in an SVG file exported by Adobe Illustrator 10.

```
<graph xmlns="&ns_graphs;"
     internalVersion="0.01"     -- Goes away for final release
     >
     <data numDataColumns="3">
         <propertyRow key="name">
             <value/>
             <value>Seattle</value>
             <value>San Jose</value>
         </propertyRow>
         <map column="0" key="name">
             <entry key="1">Sunday</entry>
             <entry key="2">Monday</entry>
             <entry key="3">Tuesday</entry>
             <entry key="4">Wednesday</entry>
             <entry key="5">Thursday</entry>
             <entry key="6">Friday</entry>
             <entry key="7">Saturday</entry>
         </map>
         <property key="units">”</property>
         <values>
             <row> <value>1</value> <value>3</value> <value>0</value> </row>
             <row> <value>2</value> <value>2</value> <value>0</value> </row>
```

```
        <row> <value>3</value> <value>0</value> <value>1</value> </row>
        <row> <value>4</value> <value>0</value> <value>0</value> </row>
        <row> <value>5</value> <value>1</value> <value>.5</value> </row>
        <row> <value>6</value> <value>3</value> <value>0</value> </row>
        <row> <value>7</value> <value>4</value> <value>0</value> </row>
    </values>
</data>

<format
    type="2D"
    width="150" height="100"
    axisLayout="cartesian"
    seriesStackingOrder="firstInFront"
    inSeriesStackingOrder="lastInFront">
    <legend position="right" order="normal"/>
    <majorAxis direction="x" orientation="+">
        <dataColumns>
            <int>0</int>
        </dataColumns>
    </majorAxis>
    <minorAxis direction="y" orientation="+">
        <dataColumns>
            <other/>
        </dataColumns>
    </minorAxis>
    <axis type="major" name="x" position="minEdge"
        ticksAndLabels="explicit"
        tickLength="short" tickPositionBase="1.5" tickSpacing="1"
        subTicks="0" labelPositionBase="1" labelSpacing="1">
        <labelString>%m:name</labelString>
    </axis>
    <axis type="minor" name="y" position="minEdge"
        ticksAndLabels="explicit"
        tickLength="short" tickPositionBase="dataMin" tickSpacing="1"
        subTicks="0" labelPositionBase="dataMin" labelSpacing="1">
        <labelString>%v%p:units</labelString>
    </axis>
    <graphDesc type="bar"
        dropShadow="none"
        majorAxisName="x" minorAxisName="y"
        columnWidth="90" clusterWidth="80"
        stacking="none" accumulation="none"
        majorRange="paddedSpan" minorRange="aiDefaultValue">
        <majorDataColumns>
            <int>0</int>
        </majorDataColumns>
        <minorDataColumns>
            <other/>
        </minorDataColumns>
        <axisNames>
```

```
                    <name>x</name>
                    <name>y</name>
                </axisNames>
                <legendString>%p:name</legendString>
            </graphDesc>
        </format>
    </graph>
```

7 Extensibility XML Elements and Attributes

7.1 About This Chapter

This chapter describes standard elements and attributes that support plug-in object extensibility for SVG files exported by Adobe Illustrator 10.

7.2 Namespace

The following is the Extensibility namespace URI:

```
http://ns.adobe.com/Extensibility/1.0/
```

7.3 DTD

The following is the DTD for the Extensibility namespace:

```
<!-- Extensibility.dtd (version 1.0) - 20010915 -->

<!-- This DTD summarizes the elements and attributes in the
     extensibility namespace exported from Adobe Illustrator 10.0..
     The namespace URI is "http://ns.adobe.com/Extensibility/1.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-EXTEND (Namespace prefix for this namespace, such as "x:")
       XMLNS-DECLARE-EXTEND (xmlns[:prefix] attribute declaration)
       (and variable XLink entities)
     and then include this DTD via reference. -->

<!-- Data types -->
<!ENTITY % URI "CDATA" >

<!ENTITY % targetRefElement "%NSPREFIX-EXTEND;targetRef" >

<!ELEMENT %targetRefElement; EMPTY >
<!ATTLIST %targetRefElement;
  %XMLNS-DECLARE-EXTEND;
  id ID #IMPLIED
  %XMLNS-DECLARE-XLINK;
  %xlinkShowAttribute; (other) #FIXED 'other'
  %xlinkActuateAttribute; (onLoad) #FIXED 'onLoad'
  %xlinkHrefAttribute; %URI; #REQUIRED >
```

Attribute definitions:

`id = "<name>"`

Standard ‘id’ attribute found in most XML grammars.

`xmlns:xlink = "http://www.w3.org/1999/xlink"`

Predefined convenience definition of the XLink namespace to that the author does not need to explicitly provide an namespace declaration for XLink attributes on this element. Because of this predefined attribute, the author can use the ‘`xlink:`’ prefix on various XLink attributes (e.g., `xlink:href`) without defining this prefix himself. This attribute should not be specified in the XML file since it is defined as #FIXED.

`xlink:show = "other"`

Standard XLink attribute whose only purpose is to alert XLink processors about the nature of the link. This attribute should not be specified in the XML file since it is defined as #FIXED.

`xlink:actuate = "onLoad"`

Standard XLink attribute whose only purpose is to alert XLink processors about the nature of the link. This attribute should not be specified in the XML file since it is defined as #FIXED.

`xlink:href = "%URI;"`

URI that indicates the referenced target. This attribute is required. If not specified, then on any attempt to perform an image replacement an error message will be generated and the result of the image replacement operation will be as if no image were rendered.

Adobe Illustrator 10 XML Elements and Attributes

8.1 About This Chapter

This chapter describes the various elements specific to Adobe Illustrator 10 and attributes that will be exported in an Adobe Illustrator namespace when SVG files are exported.

8.2 Namespace

The following is the namespace URI for Adobe Illustrator 10 XML elements and attributes:

```
http://ns.adobe.com/AdobeIllustrator/10.0/
```

8.3 DTD

The following is the DTD for the Adobe Illustrator XML elements and attributes namespace:

```
<!-- AdobeIllustrator10.dtd - 20010915 -->

<!-- This DTD summarizes the elements and attributes in the
     AdobeIllustrator10 namespace exported from Adobe Illustrator 10.0..
     The namespace URI is "http://ns.adobe.com/AdobeIlustrator/10.0/"
     This DTD is formulated under the expectation that a parent
     DTD will set up the following entities:
       NSPREFIX-AI (Namespace prefix for this namespace, such as "i:")
       XMLNS-DECLARE-AI (xmlns[:prefix] attribute declaration)
       (and variable XLink entities)
     and then include this DTD via reference. -->


<!-- Data types -->
<!ENTITY % ClassList "CDATA" >
<!ENTITY % Color "CDATA" >
<!ENTITY % Number "CDATA" >
<!ENTITY % StyleSheet "CDATA" >
<!ENTITY % URI "CDATA" >

<!-- ================================================
     Attributes to identify Adobe plugin module data types.
     Adobe has a handful of modules which plug in to various      applications, such as
the graphing module. The following attributes
     specify the namespace for the plugin module and the type of plugin      object.
Supported namespaces URIs for objectNS and possible values      for objectType:
       http://ns.adobe.com/Flows/1.0/: pointText, areaText, pathText
       http://ns.adobe.com/Graphs/1.0/: graph
```

```
      http://ns.adobe.com/ImageReplacements/1.0/: replaceable-image
     ========================================================== -->
<!ENTITY % PluginObject
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;objectNS %URI; #IMPLIED
   %NSPREFIX-AI;objectType CDATA #IMPLIED">

<!-- ========================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on SVG 'svg' elements.
       viewOrigin - two numbers (x and y)
       rulerOrigin - two numbers (x and y)
       pageBounds - four numbers (l,t,r,b)
      viewBoxInterpretation - "asCropBox" causes the viewBox to be            imported
into Illustrator as the crop rectangle.
     ========================================================== -->
<!ENTITY % IllustratorSVGElementAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;viewOrigin CDATA #IMPLIED
   %NSPREFIX-AI;rulerOrigin CDATA #IMPLIED
   %NSPREFIX-AI;pageBounds CDATA #IMPLIED"
   %NSPREFIX-AI;viewBoxInterpretation CDATA #IMPLIED>

<!-- ========================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on SVG 'g' elements.
       layer - yes or no
       dimmedPercent - one number (a percentage value without % symbol)
       color - "#" followed by two six hexadecimal digit RGB color          values
        extraneous - "self" causes the group, but not its contents,          to
be ignored on import.
     ========================================================== -->
<!ENTITY % IllustratorGElementAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;layer (yes | no) #IMPLIED
   %NSPREFIX-AI;dimmedPercent CDATA #IMPLIED
   %NSPREFIX-AI;rgbTrio CDATA #IMPLIED
   %NSPREFIX-AI;extraneous CDATA #IMPLIED" >

<!-- ========================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on SVG graphics elements.
       alphaIsShape - yes or no
       isolated - yes or no
        knockout - on or off
     ========================================================== -->
<!ENTITY % IllustratorPathElementAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;alphaIsShape (yes | no) #IMPLIED
   %NSPREFIX-AI;isolated (yes | no) #IMPLIED
   %NSPREFIX-AI;knockout (On | Off) #IMPLIED" >
```

```
<!-- ================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on SVG 'v:sampleDataSet'
          elements.
       activeDataSet - data set name string
     ================================================ -->
<!ENTITY % IllustratorSampleDataSetAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;activeDataSet CDATA #IMPLIED" >

<!-- ================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on SVG 'image' elements.
       linked - indicates whether linked or embedded
       linkRef - original file on author's editing system
       linkTransform - 2x3 matrix expressed as six numbers
       linkBBox - original bounds for image in AI coord sys (l,t,r,b)
       linkReplacement - defines how a replacement image is scaled to            fit
the original
       linkAlignment - defines how a replacement image is aligned with            the
original
     ================================================ -->
<!ENTITY % IllustratorImageElementAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;linked (yes | no) #IMPLIED
   %NSPREFIX-AI;linkRef CDATA #IMPLIED
   %NSPREFIX-AI;linkTransform CDATA #IMPLIED
   %NSPREFIX-AI;linkBBox CDATA #IMPLIED
   %NSPREFIX-AI;knockout (On | Off) #IMPLIED"
   %NSPREFIX-AI;linkReplacement (fill | fit | conform | reconform )        #IMPLIED
   %NSPREFIX-AI;linkAlignment (topleft | topmid | topright | midleft |        midmid |
midright | botleft | botmid | botright) #IMPLIED >

<!-- ================================================
     Various attributes containing Illustrator-specific
     roundtripping information that are placed on 'sampleDataSets'      elements.
     ================================================ -->
<!ENTITY % IllustratorSampleDataSetsAttributes
  "%XMLNS-DECLARE-AI;
   %NSPREFIX-AI;activeDataSet CDATA #IMPLIED" >

<!-- ================================================
    'midPointStop' element (for Adobe Illustrator roundtripping)
     ================================================ -->
<!ENTITY % midPointStopElement "%NSPREFIX-AI;midPointStop" >

<!ELEMENT %midPointStopElement; (desc|title|metadata)* >
<!ATTLIST %midPointStopElement;
  offset %Number; #IMPLIED
  stop-color %Color; #IMPLIED
  class %ClassList; #IMPLIED
  style %StyleSheet; #IMPLIED >
```

```
<!-- ================================================
    'pgf' and 'pgfRef's elements (for Adobe Illustrator roundtripping)

     Elements having to do with embedding a block of binary
     Illustrator roundtripping information at the end of the SVG file.
     The 'pgfRef' appears inside a 'foreignObject' toward the top of the      file. It
includes an href to a 'pgf' at the bottom of the file.
     ================================================ -->
<!ENTITY % pgfRefElement "%NSPREFIX-AI;pgfRef" >

<!ELEMENT %pgfRefElement; ANY >
<!ATTLIST %pgfRefElement;
  %XMLNS-DECLARE-AI;
  id ID #IMPLIED
  %XMLNS-DECLARE-XLINK;
  %xlinkShowAttribute; (other) #FIXED 'other'
  %xlinkActuateAttribute; (onLoad) #FIXED 'onLoad'
  %xlinkHrefAttribute; %URI; #REQUIRED >

<!ENTITY % pgfElement "%NSPREFIX-AI;pgf" >

<!ELEMENT %pgfElement; (#PCDATA) >
<!ATTLIST %pgfElement;
  %XMLNS-DECLARE-AI;
  id ID #IMPLIED >
```

The `pgf` element contains Illustrator `PGF` roundtripping information in the event that the user chooses the appropriate export option. The `<pgf>` element contains a base64-encoded stream of compressed data broken into chunks, with each chunk placed within a CDATA node. This chunking is done to limit the maximum contiguous sized memory block needed to load the SVG file into the DOM.

When Illustrator exports SVG with PGF included, then the graphical contents of the SVG file is enclosed in a switch. The first child of the switch is a `foreignObject`, inside of which is a `pgfRef` element which references a `pgf` element at the very end of the file. The second child of the switch is a `g` (grouping) element which contains the SVG to be rendered. On import, Illustrator encounters the switch, recognizes the `foreignObject` as an Illustrator-specific extension, reads the enclosed `PGF` and discards the following `g` element and all of its contents. When an SVG viewer is invoked, however, it will not recognize the `foreignObject` and thus will ignore the `pgfRef` and `pgf`, and will render the contents of `g` (the second child of the `foreignObject`) instead.

Note that the `pgf` element is not included directly in the foreign object. Instead it is placed at the end of the SVG document and referenced from the foreign object. This is done so that it is easy both to locate the SVG representation of the artwork and to prevent Illustrator from importing the PGF. To prevent Illustrator from importing the PGF, the recommended approach is to delete the `<pgfRef>`.