SNML Reference Manual

This is a preliminary specification. VM Labs reserves the right to make changes to any and all of the interfaces described in this document.

SNML

The Simple NUON Markup Language is used to quickly author applications using text and simple 2D graphics. This document is a reference manual. The easiest way to learn SNML is to look at the SNML Tutorial Lessons. SNML is an evolving framework for creating interactive DVD applications. This manual describes SNML Version 1.0

Valid SNML Documents

A Valid SNML Document must first of all be a well-formed XML document. This manual does not explain XML. There are a large number of books written about XML and complete documentation is available at http://www.w3.org/XML/ A quick description is at http://www.w3.org/XML/1999/XML-in-10-points. To use SNML, you do not need to know anything more about XML than you learn in the SNML Tutorial Lessons.

In addition to being a well-formed XML document, a valid SNML (version 1.0) document must use only the elements and attributes described in this manual. XML provides two mechanisms for independently validating XML documents against specific formal descriptions: Document Type Definitions and Schemas. In version 1.0, we have not provided a DTD or Schema that can be used to validate a proposed SNML document. This will be added in the future. For now you must rely upon the errors and warnings generated by the Saxon transcoding tool and the NML compiler to tell you about problems in your SNML document.

Valid SNML Elements

This section is an alphabetical listing of all the elements that can be used in a valid version 1 SNML document. For each element, an example is provided and the legal attributes and attribute values are specified. Each attribute of an element is required

unless it is labeled as optional. When an attribute is optional, a default value is provided if you do not include the attribute.

comment
A comment is any text string contained between an opening <!-- tag and a closing --> tag. Comments are only used to aid humans in reading a document. Unlike some other languages, comments are never used to sneak in instructions to processing software. A comment can occur anywhere in a document.

Attributes: None

Example:
<!-- This is a comment -->

on-click
The on-click element specifies actions to be taken when the remote-control Enter Key is pressed. The action is specified by: a single executable bob statement; or a series of executable bob statements separated by semicolons; or the name of a bob Function that is defined in a script file previously declared in this document.

Attributes: None

Usage: Use this element only as a sub-element contained in an element that can accept *focus*.

Example:
<on-click>answer.Show(true);</on-click>

on-load
The on-load element specifies actions to be taken when a page is loaded by the run-time loader. The action is specified by: a single executable bob statement; or a series of executable bob statements separated by semicolons; or the name of a bob Function that is defined in a script file previously declared in this document.

Attributes: None

Usage: Use this element only as a sub-element of a *page* element. bob statements can cause different pages to be loaded and unloaded in response to user actions and system events.

Example:
<on-load>answer.Hide(true);</on-load>

page

The page element defines an area of the TV screen to be managed, a set of elements or widgets that are active in that area, and the responses to be made to events that occur while the page or elements within the page are in *focus*.

Attributes:

width - width of the area in pixels. Range: 1 to 720;

height - height of the area in pixels. Range: 1 to 480;

back-color - Color used to fill page area each time it is drawn, prior to drawing elements contained in page. *Optional: default = linen.*

first-focus - name of the sub-element that will first be in focus when the page is loaded.

Usage: Each SNML document can have only 1 page element. It must be a sub-element of an SNML element.

Example:
<page width="720" height="480" first-focus="Question"> *pagecontent* </page>

script
The script element specifies a URL path to a file containing bob function definitions and declarations. bob is the name of the scripting language used for NUON applications.

Attributes:

src - path name to script file

Usage: A script element must be a sub-element of an snml element. There can be more than one script element contained in an snml element. A script element must occur in a document prior to any reference to functions defined in the script file.

Example:
<script src="scripts/lesson1.bbo"/>
Note: the /> is a legal shortcut in XML when an element has only properties and not sub-elements. In other words, this is treated exactly the same as
<script src="scripts/lesson1.bbo"></script>

snml
The snml element is the root element of each snml document.

Attributes: None

Usage: Each snml document should have exactly one snml element. Except for comments, all other elements should be sub-elements of the snml element.

Example:
<snml> *snml document content* </snml>

text-box
A text-box element defines an area of a page that is used to display lines of text. When a text-box is in focus, the lines can be scrolled up or down by pressing the up-arrow or down-arrow keys. When scrolling moves a line so that part of it is not within the text-box area, the line is not displayed.

Attributes:

name - The name of a particular text-box. This name is used by NUON scripts to reference the text-box. *Optional: No default.*

width - width of text-box area in pixels.

height - height of text-box area in pixels.

left - pixel offset of left boundary of text-box from left boundary of containing element.

top - pixel offset of top boundary of text-box from top boundary of containing element.

back-color - Color used to fill text-box area each time it is drawn, prior to drawing elements contained in text-box. *Optional: No default.*

Usage: A text-box element must be contained within a *page* element. A page element can contain any number of text-box elements. Each line of text is specified by a *text-line* element. A text-box element can contain any number of text-line elements.

Example:

```
<text-box width="200" height="50" left="0" top="0">
    <text-line base="30" left="0">Hello</text-line>
</text-box>
```

text-line
A text-line element defines the content, style, and position of a single line of text. It is up to the author to verify that the resulting line of text actually fits in the containing element. If the text-line does not fit, it will typically not be displayed. No automatic wrapping of text onto multiple lines is provided. The content of the text-line is all the text between the opening text-line tag and the closing text-line tag.

Attributes:

left - pixel offset of left boundary of text relative to left boundary of containing element.

base - pixel offset of text baseline relative to top boundary of containing element. The baseline of a string of text is the imaginary line on which capital letters with straight bottom edges sit.

text-style - the name of a text-style element that will be applied to this text. *Optional: Default is defaultTextStyle.*

Usage: A text-line must be contained within a *text-box* or *text-menu* element. Any number of text-line elements may be contained in a text-box or text-menu element.

Example:
<text-line left = "0" base ="50" text-style="Heading1">TITLE</text-line>

text-menu
A text-menu element defines an area of a page that is used to display a list of menu items. When a text-menu is in focus, the up-arrow and down-arrow keys can be used to select a particular item which is displayed in a special highlighted text-style. More menu-items can be defined than will fit in the text-menu. The arrow keys can be used to scroll non-visible items into view.

Attributes:
name - The name of a particular text-menu. This name is used by NUON scripts to reference the text-box. *Optional: No default.*
width - width of text-box area in pixels.
height - height of text-box area in pixels.
left - pixel offset of left boundary of text-menu from left boundary of containing element.
top - pixel offset of top boundary of text-menu from top boundary of containing element.
back-color - Color used to fill text-menu area each time it is drawn, prior to drawing elements contained in text-box. *Optional: No default.*
select-line - The number of the item to be highlighted when the text-menu is first drawn. Range - 1 to number items.
select-style - The name of the style to be used to highlight selected menu item. When an item is selected, the style specified in the text-line element for that menu item is temporarily replaced by the select-style.

Usage: A text-menu element must be contained within a *page* element. A page element can contain any number of text-box elements. Each menu-line is specified by a *text-line* element. A text-menu element can contain any number of text-line elements.

Example:
<text-menu select-line="1" select-style="hilite" width="200" height="400" left="0" top="0">
<text-line base="30" left="0" text-style="normal">item 1</text-line>
<text-line base="60" left="0" text-style="normal">item 2</text-line>
</text-menu>

text-style
A text-style element specifies the typeface, color, and other properties of displayed text.

Attributes:
name - The name of this text-style. This name is used by other elements such as text-lines to reference this particular text-style.

font - URL path to a TrueType font file. If you use the bundle tool, it will create a fonts folder containing a TrueType font called system.ttf You can put other TrueType fonts in this folder and reference them just as you reference system.ttf. *Optional: Default = " fonts/system.ttf"*

size - The nominal pixel height of a line of text in this style. However, some exotic type faces can have letters that do not fit within this size. You must proof the displayed text to verify that you are getting the appearance you want. *Optional: Default = "24"*

color - The foreground color that will be used to display the letters. *Optional: Default = "black"*

Usage: A text-style element can occur anywhere in an snml document. The name of a text-style can be referenced by text-line elements and by text-menu elements. A text-style element must appear in an snml document prior to any element that references it.

Example:
<text-style name="Heading1" font="fonts/Times.ttf" size="36" color="blue" />

Fonts

The NUON Platform uses TrueType outline fonts. One font is provided called system.ttf. You may use this font royalty free in any application that runs on the NUON platform. If you use other TrueType fonts, you should make sure you have a license from the font copyright holder to distribute it in your application.

Focus

Each snml page generally contains many elements. The behavior of the application in response to an event such as a user key press depends upon which element is currently "in focus". Each page must always designate the first element to receive focus when the page is loaded. That element retains focus until a bob changes the focus to a different element or a new page is loaded. Elements that can accept focus are called widgets. The only widgets in snml version 1.0 are text-boxes and text-menus.

Colors

Color attributes are used in many elements. The value of a color can be specified in two ways.

The easiest way to choose a color is to use one of the named colors in the following table. These names are recommended by the World Wide Web Consortium for use in XML vector graphics and in Cascading Style Sheets. See:
http://www.w3.org/TR/SVG/types.html#ColorKeywords

To exert more control over color use, you can provide a string of the form #RRGGBB, where RR is a hexadecimal number between 00 and FF representing the amount of Red in the color, and GG and BB represent the amounts of Green and Blue in the color. So color="#000000" represents the color black and color="#FFFFFF" represents the color white. Note that highly saturated colors do not look good on TV sets. To be safe, you can limit the maximum amount of any component to C0 instead of FF.

Named Colors:
aliceblue
antiquewhite
aqua
aquamarine
azure
beige
bisque
black
blanchedalmond
blue
blueviolet
brown
burlywood
cadetblue
chartreuse
chocolate
coral
cornflowerblue
cornsilk
crimson
cyan
darkblue
darkcyan
darkgoldenrod
darkgray
darkgreen
darkgrey
darkkhaki
darkmagenta
darkolivegreen
darkorange
darkorchid
darkred
darksalmon
darkseagreen
darkslateblue
darkslategray
darkslategrey

darkturquoise
darkviolet
deeppink
deepskyblue
dimgray
dimgrey
dodgerblue
firebrick
floralwhite
forestgreen
fuchsia
gainsboro
ghostwhite
gold
goldenrod
gray
grey
green
greenyellow
honeydew
hotpink
indianred
indigo
ivory
khaki
lavender
lavenderblush
lawngreen
lemonchiffon
lightblue
lightcoral
lightcyan
lightgoldenrodyellow
lightgray
lightgreen
lightgrey
lightpink
lightsalmon
lightseagreen
lightskyblue
lightslategray
lightslategrey
lightsteelblue
lightyellow
lime
limegreen

linen
magenta
maroon
mediumaquamarine
mediumblue
mediumorchid
mediumpurple
mediumseagreen
mediumslateblue
mediumspringgreen
mediumturquoise
mediumvioletred
midnightblue
mintcream
mistyrose
moccasin
navajowhite
navy
oldlace
olive
olivedrab
orange
orangered
orchid
palegoldenrod
palegreen
paleturquoise
palevioletred
papayawhip
peachpuff
peru
pink
plum
powderblue
purple
red
rosybrown
royalblue
saddlebrown
salmon
sandybrown
seagreen
seashell
sienna
silver
skyblue

slateblue
slategray
slategrey
snow
springgreen
steelblue
tan
teal
thistle
tomato
turquoise
violet
wheat
white
whitesmoke
yellow
yellowgreen