NUON Markup Language

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

NML

The NUON Markup Language, or NML for short is a simple XML based language that allows the layout of NUON pages to be described.

Valid NML Documents

A Valid NML 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.

In addition to being a well-formed XML document, a valid NML (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 NML document. This will be added in the future. For now you must rely upon the errors and warnings generated by the NML compiler to tell you about problems in your NML document.

Valid SNML Elements

This section is a listing of all the elements that can be used in a valid version 1 NML document. For each element, the legal attributes are specified.

<nml> .. </nml>

<script />

Attributes:

event=*eventId*
src=*path*

do=*script*

<textStyle />

Attributes:

name=*name*
font=*path*
size=*pointSize*
color=*color*
bgColor=*color*

<body> .. </body>

Attributes:

colorTable=*path*
onLoad=*script*
onUnload=*script*
start=*widgetName*

The <body> tag defines the top level group of widgets. Any attribute that can be used with the <group> tag can also be used in the <body> tag.

<widget />

Common attributes:

name=*name*
type=*typeName*

x=*x-position*
y=*y-position*
width=*width*
height=*height*

background=*path*
tileBackground=*boolean*
bgColor=*color*
frameColor=*color*
frameWidth=*width*
timeout=*milliseconds*
hidden={yes/true/both, no/false/none, content, background*}*

left=*widgetName*
right=*widgetName*

up=*widgetName*
down=*widgetName*

upLeft=*widgetName*
upRight=*widgetName*
downLeft=*widgetName*
downRight=*widgetName*

onFocus=*script*
onBlur=*script*
onClick=*script*
onChange=*script*

Navigate group:

navigate={leftRight, upDown, both}
immediate=*Boolean*

Align group:

align={left, center, right}
valign={top, center, bottom}

The *left*, *right*, *up*, *down*, *upLeft*, *upRight*, *downLeft* and *downRight* attributes are used for navigation among widgets. Their values should be the names of widgets to receive the focus when the corresponding navigation key is pressed. For instance:

<widget name='titleSel' down='chaptSel' />

This means that when the *titleSel* widget has focus, pressing the *down* key will cause focus to shift to the *chaptSel* widget.

<widget type='number' />

Attributes:

min=*minimumValue*
max=*maximumValue*
value=*initialValue*
textStyle=*style*
highlightTextStyle=*style*
navigate attributes

<widget type='slider' />

Attributes:

min=*minimumValue*
max=*maximumValue*
value=*initialValue*
fgColor=*color*
bgColor=*color*
navigate attributes

<widget type='select' />

Attributes:

showAll=*boolean*
vertical =*boolean*
textStyle=*style*
highlightTextStyle=*style*
choices=*choiceString*
navigate attributes
align attributes

<widget type='text' />

Attributes:

maxLength=*maxLengthInChars*
echoChar=*char*
value=*initialValue*
fromRight=*boolean*
multiline=*boolean*
textStyle=*style*
highlightTextStyle=*style*
align attributes

<widget type='textContainer' />

Attributes:

selectTextStyle=*style to be used to highlight selected line*
selectedLine=*number of line to be first selected (1 to num lines)*

<frame />
<widget type='frame' />

Attributes:

name=*name*

x=*y-position*
y=*y-position*
width=*width*
height=*height*
src=*path*

<image />
<widget type='image' />

Attributes:

name=*name*
x=*x-position*
y=*y-position*
width=*width*
height=*height*
src=*path*

<group> .. </group>
<widget type='group'> .. </widget>

Attributes:

x=*x-position*
y=*y-position*

Style values

A style value is a string consisting of attribute/value pairs separated by semi-colons.

For example:

style=’font: fonts/system.t2k; size: 14; color: #ff0000’

Style Attributes

class: *className*
font: *path*
size: *size*
color: *color*
bgColor:*color*

Script Interface

This section describes the interface between code written in Bob and the NUON pages and elements.

Every frame maintains an independent variable scope. Within this scope the following variables are defined:

parent – the scope of the parent frame
frame – the frame object itself

In addition, when a page is loaded in the frame the following additional variables are defined:

frames – an object containing a property for each child frame variable scope
widgets – an object containing a property for each widget

EventTarget

new EventTarget()

Make a new event target. Returns an EventTarget object.

EventTarget.Push()

Push an event target on the event target stack. The target on the top of the stack gets the first chance to handle an event.

EventTarget.Remove()

Remove an event target from the event target stack. It returns true if the event target is removed and false if the event target was not on the stack.

EventTarget.ObserveEvent(eventId,fcn)

Specify a function to observe an event. Each EventTarget object maintains a set of associations between event ids and observer functions. When it receives an event, it uses these associations to dispatch the event to an appropriate observer function.

EventTarget.OnEvent(eventId,fcn)

Specify a function to handle an event. Each EventTarget object maintains a set of associations between event ids and handler functions. When it receives an event, it uses these associations to dispatch the event to an appropriate handler function.

Frame

Frame.LoadPage(path)

Load a page into a frame. If a page is already loaded into the frame, it is unloaded before attempting to load the new page.

Frame.UnloadPage()

Unload the page currently loaded in a frame.

Frame.Focus()

Return the widget having the focus in a frame with a page loaded.

Frame.ObserveEvent(eventId,fcn)

Arrange for *fcn* to be called before an event with an event ID of *eventId* is handled.

Frame.OnEvent(eventId,fcn)

Arrange for *fcn* to be called when the widget receives an event with an event ID of *eventId.*

Widget

Widget.ObserveEvent(eventId,fcn)

Arrange for *fcn* to be called before an event with an event ID of *eventId* is handled.

Widget.OnEvent(eventId,fcn)

Arrange for *fcn* to be called when the widget receives an event with an event ID of *eventId.*

Widget.SetFocus()

Set the focus to this widget.

Widget.AddElement()

Add an element to a select widget.

Widget.Show()
Widget.Show(showBackgroundP)

Show a widget.

Widget.Hide()

Widget.Hide(hideBackgroundP)

Hide a widget.

Widget.ShowBackground()

Show the background of a widget.

Widget.HideBackground()

Hide the background of a widget.

Widget.x
Widget.y
Widget.width
Widget.height

The coordinates and size.

Widget.value

The current value.

Widget.minValue

The minimum value.

Widget.maxValue

The maximum value.

Widget.style

The normal text style.

Widget.highlightStyle

The highlight text style.

Widget.bgColor

The background color.

Widget.background

The background image.

Widget.highlighted

The highlight state.

Widget.src

Widget.highlightSrc

TextStyle

new TextStyle(styleParameterObject)

Make a new TextStyle object. The styleParameterObject is an object containing the following properties:

font
size
color
bgColor

Style.CacheGlyphs(str=*defaultStr*);

The default string consists of the 26 uppercase and 26 lowercase letters and the digits.

Style.CacheGlyphsFromFile(path);

Timer

new Timer(fcn,delay=0,repeatP=false)

Make a new Timer object. If the delay (in milliseconds) is greater than zero, the timer is also set.

Timer.Set(delay,repeatP=false)

Set the timer to go off in a specified number of milliseconds.

Timer.Clear()

Clear a Timer.

Timer.timeRemaining

The number of milliseconds remaining before the timer goes off.