



Portal Developers Guide

Rev 3.0.0

19th January 2006

Portal Developers Manual

This document has been written to allow developers to create their own portal windows using the IntraNomic branding and standard coding routines.

The IntraNomic web directory (drive:\inetpub\wwwRoot\IntraNomic) includes a “Developer Examples\Active Portal” directory, which combined with this document, should allow you to create new IntraNomic portal windows.

The directory “Developer Examples/Active Portal/Template” contains an ASP page called “template.asp”, which should be used as the basis for your new IntraNomic window. The template page includes the initial ASP code, includes files and JavaScript to produce a completely blank IntraNomic portal window.

The following document will then give HTML and JavaScript code examples to enhance your page.

ASP Functionality

ASP Title

Adds the window title to the web page.

Syntax

`pageTitle = [text string]`

Example

`pageTitle = "My Window"`

ASP Event Variables

ASP events can be setup to call JavaScript functions.

Syntax

```
pageLoad = [function call text string]
```

Example

The following example assigns the **load** function to the **onload** event of the HTML pages body tag.

```
pageLoad = "load();"
```

Events

JavaScript functions can be assigned to the following list of ASP variables. The corresponding page event is displayed in the description.

pageLoad	–	Fired onload of the window (onload)
pageOnClick	–	Fired when the window BODY is clicked (onclick)
pageKeydown	–	Fired when the keyboard is pressed (onkeydown)
pageFocus	–	Fired when window receives the focus (onfocus)
pageBlur	–	Fired when window loses focus (onblur)
pageMouseMove	–	Fired when the mouse moves over the window (onmousemove)
pageResize	–	Fired on resize of the window (onresize)
pageBeforeUnload	–	Fired before unload allows a code to stop a window from closing if required (onbeforeunload)
pageUnload	–	Fired when the window is unloaded (onunload)

Remarks

These events are ASP variables, which are then assigned to the HTML body tag events when the page is loaded from the web server. The events are global to the page not to a specific element within the HTML body.

JavaScript Functionality

Copy to Clipboard

An IntraNomic function that copies text directly to the users clipboard.

Syntax

```
copyToClipboard(text string);
```

Example

```
copyToClipboard("This is the text I wish to copy");
```

or

```
function copyButton(strPassText) {  
    copyToClipboard(strPassText);  
    alert("The text has been copied to the clipboard");  
}
```

Remarks

Once copied to the contents of the clipboard can be pasted into other applications. Each time the function is called the previous text is overwritten.

F-Key Assignments

IntraNomic allows F3 – F11 to be assigned to specific JavaScript functions written by the developer.

Syntax

`g_strF3Accellerator = [function call text string]`

Examples

```
g_strF3Accellerator="processFKey(3);
```

`g_strF3Accellerator` is a global variable, which allows a function to be assigned to the relevant F-key in this case "F3". The function called in this example is `processFKey()`, which accepts one parameter.

```
g_strF9Accellerator = "window.close()";
```

When the "F9" key is pressed the current window will be closed.

Remarks

The F-Keys work against the current active window. F-Keys are global to the entire page and cannot be assigned to a sub section of a window.

F-Key Cancel Assignment

Allows the removal of an F-Key assignment.

Syntax

```
cancelKeyAccell(text string);
```

Parameters

text string	–	“F3” to “F11”
-------------	---	---------------

Example

```
cancelKeyAccell("F3");
```

Remarks

Allows a developer to dynamically remove an F-Key assignment through code.

Hide HTML Element

Allows the developer to hide a HTML element or block of HTML.

Syntax

```
hideLayer(object);
```

Parameters

object	–	The HTML element you wish to hide. The element is passed to the function as an object.
--------	---	--

Examples

This example sets up an element “divContent”, which will be displayed as the page loads.

```
<div id="divContent">This is my content area that I wish to hide</div>
```

Your code could then use the hideLayer function to hide the text when a certain event occurs on the page:

```
hideLayer(divContent);
```

Remarks

The HTML element passed to hideLayer should never include quotes or double-quotes around the element identity i.e. “divContent” or ‘divContent’.

Show HTML Element

Allows the developer to show a HTML element or block of HTML.

Syntax

```
showLayer(object);
```

Parameters

object	–	The HTML element you wish to show. The element is passed to the function as an object.
--------	---	--

Examples

This example sets up an element “divContent”, which is hidden when the page loads.

```
<div id="divContent" style="display:none;">This is my content area that I wish to hide</div>
```

Your code could then use the showLayer function to display the text when a certain event occurs on the window:

```
showLayer(divContent);
```

Remarks

The HTML element passed to showLayer should never include quotes or double-quotes around the element identity i.e. “divContent” or ‘divContent’.

Creating Branded Buttons

Allows the creation of branded buttons in two sizes small and large.



Syntax

`makeImgButton(buttonID, text, action, size, width, state, placeholder, rootPath);`

Parameters

buttonID	-	The button ID which must be unique to the current HTML page
text	-	The display text i.e. "Show" or "Hide" etc
action	-	The JavaScript function called when the button is clicked
size	-	Integer value: large = 1, small = 2
width	-	Button width default 0 will size to inner text width
state	-	Initial button disabled state i.e. Enabled = false, Disabled = true
placeholder	-	The element where the button is created. If left blank the button is created inline and placed at the point where the code is run. If the code is run from a function this property must be added to tell the function where to write the button HTML.
rootPath	-	Required to find the actual path to the button images. Normally set to the web root i.e. "/IntraNomic/"

Examples

This example shows the inline code where the button is created as the page is loaded.

```
<table cellspacing=0 cellpadding=0 border=0>
  <tr>
    <td id=tdButton style="padding-left:10px;">
      <script language='JavaScript'>

        makeImgButton("btnShowLarge","Show","showLayer(divContent);",1,0,false,"","/IntraNomic/");

      </script>
    </td>
  </tr>
</table>
```

Creating Branded Buttons continued.

Examples continued.

This example shows a JavaScript function, which creates a button using function. Notice that the HTML element "tdButton" is added to tell makeImgButton where to place the button within the page.

```
function createButton() {  
    makeImgButton("btnShowLarge","Show", "showLayer(tdLargeText);",1,0,false," tdButton","/IntraNomic/");  
}
```

Enable / Disable Branded Button

Once a button has been created the button state can be toggled between enabled and disabled.

Syntax

```
toggleImgButton(buttonID, size, state, pathToImages);
```

Parameters

buttonID	–	The buttons HTML identity
size	–	Integer value – large = 1, small = 2
state	–	Toggle State: enabled = 3 and disabled = 4
pathToImages	–	Required to find the actual path to the button images.

Examples

This example shows a large button being enabled:

```
function enableButton(strPassButtonID) {  
    toggleImgButton(strPassButtonID, 1, 3, "/IntraNomic/");  
}
```

This example shows a large button being disabled:

```
function disableButton(strPassButtonID) {  
    toggleImgButton(strPassButtonID, 1, 4, "/IntraNomic/");  
}
```

Remarks

Disabled buttons are displayed in dark grey and the action assigned to the button will not be processed when the button is clicked.

The colour of an enabled button is determined by the IntraNomic branding i.e. Blue, Green, Brown, Purple or Red.

Executing External Applications

Function used to execute programs stored on your network. It requires the file path and any parameters required to execute the application.

Syntax

```
executeProcess(actualPath, parameters, startInPath);
```

Parameters

actualPath	-	Full execution path of the application
parameters	-	Execution parameters which may include spaces
startInPath	-	Certain applications require the start in folder path

Examples

A simple example:

```
executeProcess("c:/test.exe", "", "");
```

An example using the startInPath:

```
executeProcess ("c:/test folder/lee/test.exe", "", "C:/test folder/lee");
```

A complex example, which executes MS-Word to open a specific document:

```
executeProcess("C:/Program Files/Microsoft Office/Office/WINWORD.EXE",  
"-o \"C:/GCC Intranet.doc\"", "");
```

Remarks

You may also see the backslashes on these examples this is a control character in JavaScript that allows the preceding character to be used within the statement i.e. to display double-quotes around the text - "\"test.exe\"" would display "test.exe".

File Size Formatting

Takes a number of bytes and returns a formatted string i.e. **1.7 GB**, **1.0 MB**, **30.5 KB** etc.

Syntax

```
formatFileSize(size);
```

Parameters

size	–	Size of the file in bytes
------	---	---------------------------

Examples

This example returns the file size in Kbytes:

```
var strReturn = formatFileSize(1500);
```

strReturn will be returned as "1.4 KB".

Remarks

A simple function that allows for a standard display format for all file sizes.

Retrieve Windows Registry Key

The registry can be accessed using the IntraNomic custom built "Utility DLL". This will allow a developer to access specific client side information, which may differ depending on the PC that accessed the window.

Syntax

```
g_objSTSIU.GetRegStringValue(keyNode, regKey, subKey);
```

Parameters

keyNode	–	Initial Node Root
regKey	–	Path to subKey Node
SubKey	–	Returns the specific value of the specified subKey

Examples

This example retrieves the Desktop path from the registry

```
function testGetRegistryKey() {  
  
    var strKeyNode="HKEY_CURRENT_USER";  
    var strRegKey="software\\microsoft\\windows\\currentversion\\explorer\\shell folders";  
    var strSubKey="Desktop"  
  
    return g_objSTSIU.GetRegStringValue(strKeyNode, strRegKey, strSubKey);  
}
```

Remarks

The backslash is a control character in JavaScript so this example has to double up the backslash to display it once i.e. "software\\microsoft" would display "software\microsoft".

Retrieve “My Documents” Folder

Uses the IntraNomic custom built “Utility DLL” to retrieve the client PC’s “My Documents” folder path.

Syntax

Returns a text string of the current PC’s “My Documents” folder path.

```
var strWindowsPath = g_objSTSIU.GetWindowsFolder();
```

Retrieve Windows Folder

Uses the IntraNomic custom built “Utility DLL” to retrieve the client PC’s “Windows” folder path.

Syntax

Returns a text string of the current PC’s “Windows” folder path.

```
var strWindowsPath = g_objSTSIU.GetWindowsFolder();
```

IntraNomic Branding Functionality

If a developer wishes to build a window that uses the current branding the following functions may be used.

Add / Remove Scrollbars

By default the IntraNomic portal windows does not display scrollbars. This JavaScript command should be added when the window is loaded.

Syntax

```
document.body.scroll="yes";
```

or

```
document.body.scroll="no";
```

Allow Right Mouse Click

By default an IntraNomic portal window stops the right mouse click, which displays the window browser menu. This JavaScript command should be added when the window is loaded.

Syntax

```
document.body.oncontextmenu="";
```

Remarks

The command removes the internal function that suppresses the right mouse click.

Suppress Right Mouse Click

This functionality is enabled by default.

Syntax

If the functionality has been removed this code example will reset the default.

```
document.body.oncontextmenu="event.returnValue=false;event.cancelBubble=true;";
```

Background and Scrollbar Branding

Displays the page background and scrollbars using the current IntraNomic branding. This function call should be added when the window is loaded.

Syntax

```
windowBranding();
```

Remarks

The default IntraNomic background uses a gradient effect for the selected branding colour.

Section Headings

The standard IntraNomic section headings consist of the required text followed by a line delimiter, which allows a clear break between HTML sections.

Section Heading

Syntax

```
createHTMLTitle(title, colour, size);
```

Parameters

title	–	The text displayed on the header
colour	–	Colour of the section heading i.e. #000099
size	–	The font-size in pixels

Examples

In the following example the IntraNomic E-Learning section heading is displayed using the primary branding colour and a font size of "11px".

```
divTitle.innerHTML = createHTMLTitle("IntraNomic E-Learning", g_objBranding.primaryColor, "11");
```

IntraNomic E-Learning

Remarks

The example above sets up a placeholder element, which is then populated by the createHTMLTitle function when the HTML page is loaded from the web server.

Standard Fonts

Several standard font classes have been made available to the developer all of which use **Arial** as the base font family.

Classes

standard8	–	Add "class=standard8" to HTML element - [normal 8pt Arial]
standard8bold	–	Add "class=standard8bold" to HTML element - [bold 8pt Arial]
standard10	–	Add "class=standard10" to HTML element - [normal 10pt Arial]
standard10bold	–	Add "class=standard10bold" to HTML element - [normal 10pt Arial]

Syntax

This example uses the font class "standard8":

```
<table cellspacing=10 cellpadding=0 border=0>  
  <tr>  
    <td class=standard8>Test Font</td>  
  </tr>  
</table>
```

Branding Colours

When creating a web page using the IntraNomic development template the developer can use the following branding colour properties:

This example shows the default branding of [blue](#)

Property		Colour
g_objBranding.primaryColor	–	000099
g_objBranding.secondaryColor	–	6666CC
g_objBranding.thirdColor	–	7979F1
g_objBranding.splashColor	–	E2E0F8
g_objBranding.messageGradient1	–	95B9F5
g_objBranding.messageGradient2	–	D5E6FA
g_objBranding.menuRollover1	–	BBBBFF
g_objBranding.menuRollover2	–	3399FF
g_objBranding.headerBackground	–	AEAEEB
g_objBranding.detailGradientDark	–	DCDCFD
g_objBranding.scrollbarBase	–	ECECFD
g_objBranding.scrollbar3Dlight	–	EEEEEE
g_objBranding.scrollbarArrow	–	5454BA
g_objBranding.scrollbarShadow	–	CCCCCC
g_objBranding.highLightColor	–	FFD700

By using the branding colours the window will automatically co-ordinate with the current client's IntraNomic settings. Other base branding colours are Green, Brown, Purple and Red.

Syntax

This code will set the table cell (“tdCol1”) to the primary branding colour. Also note that the branding property does not include the “#”, which is required to display the specified colour.

```
tdCol1.style.backgroundColor="#" + g_objBranding.primaryColor;
```