
GEPS Intranet Menu Standards
Using the XML Menuing System
Version: 1.0

Published: July 19, 2002

Author: UI Standards Team



GE Power Systems
e-Technology

© 2001 **General Electric Company**. All rights reserved.

This document is GE proprietary and confidential.

Contents

PREFACE	3
Purpose, Scope, and Audience	3
Additional Resources	3
Conventions	3
Menu Standards	5
When to Use Menus	5
Hierarchy and Scope	5
Consistency	6
Colors, Fonts, Borders, Roll-overs, and Other Effects	6
Borders	10
Full Width/Height Background Bar	11
Menu Names	13
Menu Coding	13
Configuring Basic Menus	17
Technical Syntax Issues	19
Using Externally Triggered Menus	21
Configuring the Menus for External Triggers	21
Defining the External Menu Triggers in the Page	22
Using Multiple MenuSystem Elements	25
Assigning Menus Dynamically	27
XML Menu Configuration Markup	31
Elements	31
Attributes: A-E	32
Attributes: D-E	36
Attributes: F-M	37
Attributes: H-M	40
Attributes: N-Z	42
APPENDIX A RELEASE NOTES	45
Version 1.5, 1.6, and 1.7	45
Version 1.4	46
Version 1.3	47
APPENDIX B TECHNICAL IMPLEMENTATION	49
GLOSSARY	51

Examples

Example 1. Simple Vertical Background Bar	12
Example 2. Placing Menus Directly Under the Global Masthead	13
Example 3. Multi-Line JavaScript Variable for Menus	20
Example 4. Menu Configuration for External Triggers	21
Example 5. External Menu Trigger Code in the Page	22
Example 6. Multiple Menu Systems	25
Example 7. JSP Logic for Dynamic Menus	27
Example 8. Sample Blue Menu Configuration	28
Example 9. Sample Silver Menu Configuration	28
Example 10. Sample Gold Menu Configuration	29
Example 11. Menu System JavaScript Files	50

Tables

Table 1. Standard Colors, Fonts and Sizes	7
Table 2. Sample Menu Function Access by User Level	28
Table 3. Valid Attributes for menuSystem and menuItem	31

Preface

Purpose, Scope, and Audience

This document is one in a set of three defining user interface standards for Intranet web development. It also contains implementation instructions for the XML Menu System, the application that GEPS uses to handle menuing for web sites and applications.

Web sites that should be deployed in the Epicentric Portal Server use the Epicentric menuing system rather than the XML Menu System. The menu standards in this document still apply. See Epicentric documentation for implementation instructions.

Please send any comments, questions, or suggestions to GEPSBizStds@ps.ge.com.

Interested readers may include developers, designers, analysts, technical writers, and content owners of any web site or application that is deployed from the Inside GEPS portal. Because of the widely different areas of interest and expertise, some content may be highly technical.

Additional Resources

Other standards documents in this set or those that may be of interest include:

- [GEPS Intranet User Interface Standards](#): for standards and implementation instructions on creating menus for web sites or applications using the GEPS Global User Interface. This document also contains detailed reference information.
- [GEPS Global UI for Epicentric](#): for implementation instructions on creating web sites and menus with the GEPS Global UI in the Epicentric Portal Server.
- [GEPS UI Standard JavaScript Library](#): for reference information on the functions available within the GEPS Standard JavaScript Library. This library is part of the Global UI and thus available for use in any page.
- [Development Standards](#): for the technology stack and back-end development standards.
- [Online Help Standards](#): for standards on writing and developing online help for applications.

Conventions

Several typographic conventions are used within this document:

Code examples

Examples of markup or other code are shown in this font:

```
var += xHeight + 50;
```

Variable content

Examples of URLs, commands, or other code may contain variable content that you supply based on your context. This variable content is represented by a placeholder that appears in italics, such as:

/web-app-root/html

Variables

JSP variables are shown inside braces, such as {variableA}

Menu Standards

Menus are one of the primary 'paradigms' used to present a hierarchy of navigation or other interaction choices to users. Thus, menus always have at least two levels of choices -- the parent or primary level and one or more child levels. Choices may represent navigation or, with applications, they may represent actions that users can take.

Menus are only one navigation mode that web sites and applications can use. See the [GEPS Intranet User Interface Standards](#) for information on other valid navigation modes, JSP variables, and related standards such as labeling for menu items.

When to Use Menus

All web sites should use menus for navigation within the web site. Global navigation is supplied in the Inside GEPS menu.

Applications may also use menus for their functional flow where the design criteria fit the functional requirements of the application.

Keep the following criteria and limitations for menus in mind:

- Each choice (menu item) must be independent of other choices -- there is no prescribed order or chronology of choices required.
- Users should have complete discretion over the next choice. Thus, there are no unacceptable consequences if users jump across menus or levels erratically.
- The choices available to users can be organized in a hierarchy (the tree) that is useful to assist them in discovering choices.
- The user should continue to have access to the full hierarchy of choices from any level.

Note: Applications sometimes need to limit access to global menus or those of the web sites that access the application. See the Application Options to Control Global Navigation/Shutdown section in the [GEPS Intranet User Interface Standards](#) for specific rules and implementations.

Hierarchy and Scope

The number of top-level menu items should fit naturally within the minimum space defined by the menu layout without causing users to scroll. This means that items on horizontal menu bars should fit in the minimum width (800 pixels). Intranet sites are encouraged to fit menu

items in vertical menu bars within the minimum height (600 pixels) wherever possible to minimize scrolling. This ensures that users have easy access to the global and site-specific navigation.

Menu levels (hierarchy) should be limited to three levels, with two levels strongly preferred wherever possible. Users cannot easily navigate with the mouse more than three levels because of the dexterity requirements.

Note: If applications use frames, menus may have only **one** level. Drop down menus do not perform reliably when frames are present, so only one level is supported.

Web sites and applications that use menus **must** follow these basic rules:

- Choose either vertical or horizontal menus as the primary, site-specific menus and make them consistent throughout the site or application. Applications that may move to the Extranet should use horizontal menus as vertical menus are not allowed in the Extranet.
- If the menu hierarchy absolutely requires more than three levels, use a second set of menus for deeper areas of the site, but include a "home" menu item that leads back to higher areas of the site. Or use hierarchical 'bread crumb' trails to give users instant access to higher levels.
- If some specific pages or areas of the site need additional menus, use the opposite orientation and clearly label or indicate that this is local to the page. So, if vertical menus are the primary site menus, use horizontal menus for page-specific or local menus.

The Global UI includes a global menu. Each web site then may have regional navigation in up to three levels of menus that are referred to as primary, secondary and tertiary navigation.

Consistency

Menus must be consistent with:

- Naming standards.
- Spatial relationships with respect to other elements on the page, especially as users expand or contract the browser window (only up to the minimum size).
- Other graphical design aspects, such as color.
- Use of any hinting techniques. This applies to passive techniques such as using ellipses to indicate more choices or to interactive techniques such as active highlighting.

Colors, Fonts, Borders, Roll-overs, and Other Effects

The XML Menu System implements very straight-forward menus, and this is the preferred look. This is lightweight, which helps performance,

and easily maintainable which makes upgrades to new look and feel standards simpler.

Table 1. Standard Colors, Fonts and Sizes

Menu Property (XML attribute)	Valid Values for Extranet/Internet/Intranet
Font (fontFamily)	Verdana
Font Weight (fontWeight)	Don't specify a weight. This system uses the default value of medium.
Text Color (fontColor)	<ul style="list-style-type: none"> • Horizontal menus: #000000 • Vertical menus: #003399
Highlighted Text Color (fontOnColor)	<ul style="list-style-type: none"> • Horizontal menus: #666666 • Vertical menus: #eeeeee <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: To ensure a simple roll-over effect, this color must be different from the base text color in fontColor.</p> </div>
Font Size (fontSize)	11px (~8pt)
Outside Borders (boxBorder and boxBorderColor)	<ul style="list-style-type: none"> • Horizontal menus: do not set • Vertical menus: <ul style="list-style-type: none"> • boxBorder = 0 for the top level (set on menuSystem element) and 1 for subsequent levels (set on first-level menuitem elements) • boxBorderColor = #003399 (set on first-level menuitem elements)

Table 1. Standard Colors, Fonts and Sizes (continued)

Menu Property (XML attribute)	Valid Values for Extranet/InternetIntranet
Inter-Item Borders (intBorder and backColor)	<ul style="list-style-type: none"> • <input type="checkbox"/> Horizontal menus: do not set intBorder. Set backColor = #eaeaea. • <input type="checkbox"/> Vertical menus: do not set intBorder. Do not set backColor on the top level (menuSystem), but is set to #a3a3ed on the next level (top-level menuitem elements).
Background (bgOffColor)	<ul style="list-style-type: none"> • <input type="checkbox"/> Horizontal menus: <ul style="list-style-type: none"> • <input type="checkbox"/> Set on top level (menuSystem) with a null value (""). The Global UI supplies a tiled graphic with a gray background and some shading. • <input type="checkbox"/> Set to #eaeaea on first-level menuitem elements. • <input type="checkbox"/> Vertical menus: #e7e7ff set at the top level (menuSystem) <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: Background colors are required to help ensure that menus are distinct from simple content. This attribute is only allowed to be null ("") for horizontal menus that use the standard horizontal menu bar (the color is supplied by the Global UI).</p> </div>

Table 1. Standard Colors, Fonts and Sizes (continued)

Menu Property (XML attribute)	Valid Values for Extranet/Internet/Intranet
Highlighted Background (bgOnColor)	<ul style="list-style-type: none"> • <input type="checkbox"/> Horizontal menus: <ul style="list-style-type: none"> • <input type="checkbox"/> Set on top level (menuSystem) with a null value (""). The Global UI supplies a tiled graphic with a gray background and some shading. • <input type="checkbox"/> Set to #ffffff on first level menuitem elements. • <input type="checkbox"/> Vertical menus: #003399 set at the top level (menuSystem) <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: This value must be different from the base background to ensure a simple roll-over effect for menus. This attribute is only allowed to be null ("") for horizontal menus that use the standard horizontal menu bar (the color is supplied by the Global UI).</p> </div>

Table 1. Standard Colors, Fonts and Sizes (continued)

Menu Property (XML attribute)	Valid Values for Extranet/Internet/Intranet
Submenu Hint Graphic (arrowHref)	<ul style="list-style-type: none"> • Horizontal menus: <ul style="list-style-type: none"> • Top level items: use the {topLevelArrow} JavaScript variable. See the Configuring Basic Menus section on page 17 for the exact syntax. • Second level items: use the {subLevelArrow} JavaScript variable. See the Configuring Basic Menus section on page 17 for the exact syntax. • Vertical menus: arrowBlue.gif (supplied in /examples/intranet of the Global UI). Set this on the first-level menuitem elements
Submenu Hint Alignment/Padding (arrowPadding)	<ul style="list-style-type: none"> • Horizontal menus: <ul style="list-style-type: none"> • Set to 10 on menuSystem for the top-level menu items • Do not set on first-level menuitem elements (leaves arrows right-aligned for second-level menu items). For Netscape compatability, however, add the {NSarrowPad} variable and make sure that this variable is defined in your menu configuration file(s) (menuConfig.js, etc.). • Vertical menus: do not set (leaves arrows right-aligned everywhere).

The system supports a very simple roll-over effect (basic highlighting with color). Fancier effects or rollovers using graphics should be avoided.

Borders

Several attributes control the width and color of borders. The "exterior" border surrounds an entire level of the menu, except that borders do **not** affect the first level of a horizontal menu. The "interior" or "inter-item"

borders define a consistent grid around each item within a level, as shown here.

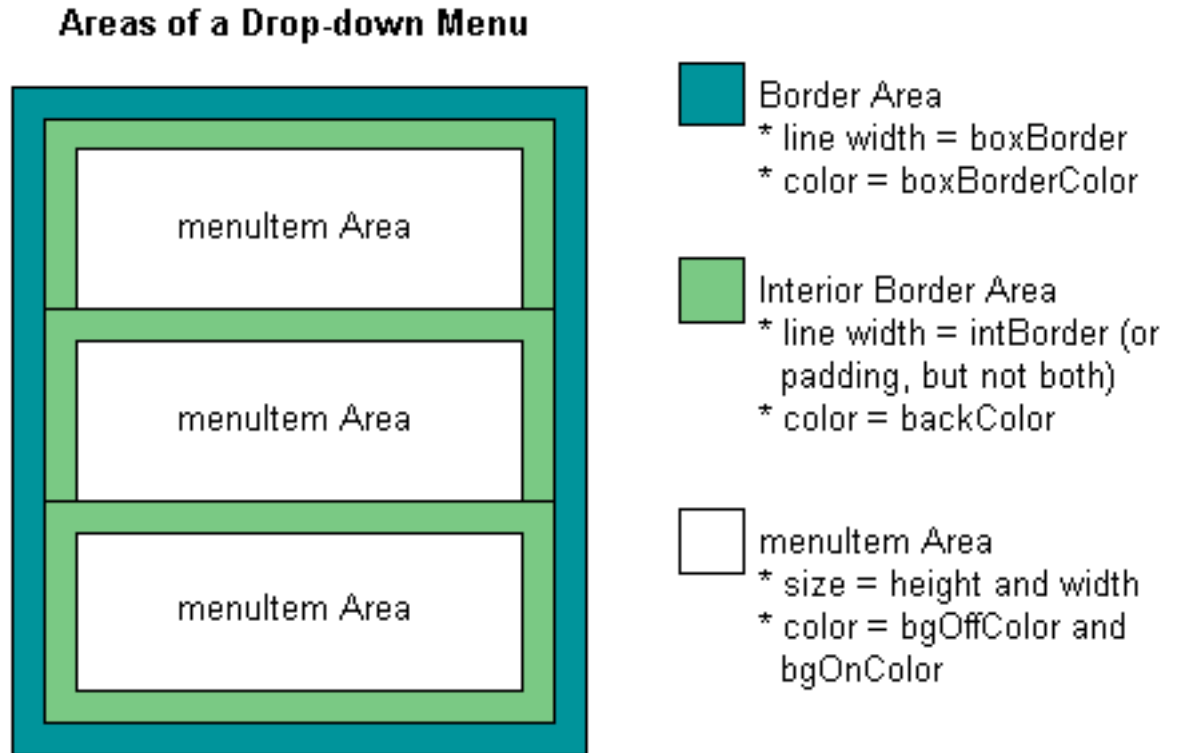


Figure 1. Border and Menu Item Areas

If you also use 'background bars' behind the menu, you can blend borders into the background bar or make the menu area distinct using border colors.

Full Width/Height Background Bar

The XML Menuing System does not extend backgrounds beyond the width or height defined for a menu system. If you want a 'bar' that extends (and contracts) the full width of the page, for horizontal menus, or the full height of the page, for vertical menus, each page must supply this separately and you must configure the menu to float over the bar.

Note: If the application or web site uses horizontal menus, you can configure the site to use the default horizontal bar that comes with the Global UI using the {mastheadSiteMenus} JSP variable. This bar uses a grey background and a drop shadow. See the [GEPS Intranet User Interface Standards](#) for more information.

Supplying a background bar simply consists of using a table and making one row or one column supply the background. You can use CSS styles or use graphics to provide a more sophisticated look to the background.

Example 1. Simple Vertical Background Bar

```
<!-- violet background -->
<table cellpadding="0" cellspacing="0" border="0" bgcolor="#A3A3ED">
<!-- thin blue top border -->
  <tr>
    <td height="1" bgcolor="#FFFFFF"></td>
    <td colspan="3" height="1" bgcolor="#003399">
      </td>
  </tr>
<!-- this blue right/left borders -->
  <tr>
    <td width="10" bgcolor="#FFFFFF">&nbsp;</td>
    <td width="1" bgcolor="#003399">
      
    </td>
    <td height="400" width="185" valign="bottom" align="center">&nbsp;</td>
    <td width="1" bgcolor="#003399">
      
    </td>
  </tr>
<!-- thin blue bottom border -->
  <tr>
    <td height="1" bgcolor="#FFFFFF"></td>
    <td colspan="3" height="1" bgcolor="#003399">
      </td>
  </tr>
</table>
```

Positioning the first-level of the menu over the background bar is controlled with the `startX` and `startY` attributes. You may use the `{navMenuY}` JavaScript variable as the value of `startY` to snug either horizontal or vertical menus directly under the masthead. An example of how this variable is used within the menu configuration information is shown here:

Example 2. Placing Menus Directly Under the Global Masthead

```
theXMLIs = theXMLIs +  
'<menuSystem startX="8" startY='+navMenuY+' width="175" ...'
```

See the Configuring Basic Menus section on page 17 for more details on specifying menu configuration information.

Menu Names

Menu labels should follow the common naming standards, especially those relating to space-limited GUI elements. See the Common Naming Standards section in the [GEPS Intranet User Interface Standards](#) for details. Menu items should not wrap to two lines as it makes them more difficult to read and more difficult to determine where the items begin and end.

Menu Coding

The XML Menueing System provides a standard, simple way to implement pull-down menus for pages in the GEPS Global UI. This menueing application is designed to work in Microsoft Internet Explorer 5.0 and higher and Netscape Communicator 4.7 (Communicator 4.7 or Navigator 4.08) and higher.

<p>Note: Web sites that should be deployed in the Epicentric Portal Server use the Epicentric menueing system. See Epicentric documentation for implementation and coding instructions.</p>
--

To define your menus, you must complete these steps:

1. Turn on menueing in the applicationSettings.jspi file by setting the {siteMenus} variable to true.
2. If your web site or application is using vertical menus or a custom background for horizontal menus, set the {mastheadSiteMenus} variable to false.

Make sure to include any background bars needed for your menus in each page of the web site or application. See the Full Width/Height Background Bar section on page 11 for specific information.

3. Copy the menuConfig.js file from the standard distribution (see the /examples/intranet directory) to the directory tree for your web site or

application. Normally, the menuConfig.js file should be placed in the directory where the JSP pages for the web site or application reside.

Note: The menuConfig.js in the standard distribution is configured as a horizontal menu that works with the standard horizontal background bar supplied in the masthead.

Copy the leftNavConfig.js file from the standard distribution instead to use as a starting point for vertical menus. You may change the name of this file to menuConfig.js to automatically incorporate this into your web site or application.

You can change the name of this file and you can move it to a subdirectory for the web site or application. If you rename it or move it, you must also change the {menuFileURL} variable in applicationSettings.jspi to the correct relative path and file name.

4. Enter the XML configuration information needed to define the content and behavior of your menus. You may use four common menu configurations:

- **Basic Menus:** these menus determine the look and feel, labels, and behavior of the menus entirely from XML configuration information. No additional elements or code are needed on the web page.

These menus are considered "internally triggered" menus. The two biggest advantages to using internally triggered menus are the simplicity of the configuration and the total control of the menu position regardless of window resizing or fluid HTML layouts. See the Configuring Basic Menus section on page 17 for coding instructions.

The primary disadvantage of internally triggered menus is that some more sophisticated design options, especially those involving graphics, are not possible.

- **Using Externally Triggered Menus section:** these menus have the first level of menuitem elements coded directly into the HTML page rather than being rendered dynamically from XML configuration information. All menu options are still rendered dynamically from the XML configuration information.

Use externally triggered menus if:

- The web site or application is Intranet-only. Current Extranet/Internet standards for menu look and feel require internally-triggered menus.
- Top-level menu items need images or other XHTML elements along with the label
- Current menu attributes do not support other visual characteristics for your menus

Externally triggered menus can support greater design options, but they also require significantly more setup and maintenance as menu configuration is now controlled by both the individual pages and the XML files. Architecturally, externally triggered menus are **deprecated**.

- **Using Multiple MenuSystem Elements section:** multiple internally-triggered or externally-triggered menus may be used in Intranet web sites or applications. Use multiple menu systems to handle deep hierarchical structures or to provide local, page-specific menus. Multiple menus are not allowed on the Extranet.
- **Assigning Menus Dynamically section:** this can be set up using the {menuFileURL} variable. It should only be used with internally-triggered menus.

Dynamic menus can contain different options for specific types of users. Or they can be configured to contain options that are relevant to the current context or state of an application.

Configuring Basic Menus

Basic menus define all the configuration information for menus in the menuConfig.js file as one menu system. The most basic content you need to enter includes:

1. A menuSystem element surrounds the full tree of the menu. The example shown below covers the basic information needed for menuSystem.

```
var theXMLIs='+  
  '<menuSystem useInternalTriggers="yes" layerTriggerPrefix="zzz">'+  
  your menu items and tree will go here  
  '</menuSystem>';
```

The value to use in layerTriggerPrefix must simply be a unique string that identifies the layers for this menu. It must be different from any other layer ID for the pages of your web site or application.

Other attributes that you can set on menuSystem to control the look of the menu include:

- Placement on the page: startX, startY, and orientation. Web sites and applications should consistently choose one menu orientation (horizontal or vertical) throughout the site or application. Menus with the opposite orientation are only allowed locally on pages and must clearly indicate that they are "local." Applications that may move to the Extranet must use a horizontal orientation if they use menus.

Set startY="' +navMenuY+' " to insure that menus (either horizontal or vertical) butt up against the bottom of the standard masthead.

- Size: height and width.

If the web site or application uses horizontal menus with the standard background supplied in the global masthead, The height attribute should be set to 18.

If the width of each menu item at the top level of horizontal menus should be different, use the localWidth attribute on each top-level menuItem element.

- Fonts: fontColor, fontFamily, fontOnColor, fontSize, and fontWeight. See Table 1 on page 7 for the correct values to match standards.

- Submenu Hinting: arrowHref and arrowPadding define graphic hinting for submenus.

Note: The arrowHref attribute may use one of two JavaScript variables on horizontal menus to use the standard graphics for submenu hinting depending on the menu level. Padding can also vary:

- Set on menuSystem to define submenu hinting for top-level menu items: arrowHref="'+topLevelArrow+'" arrowPadding="10"
- Set on first-level menuItem elements to provide Netscape compatibility and to define submenu hinting for second-level menu items: arrowHref="'+subLevelArrow+NSarrowPad+'" and **do not** add padding.

One additional JavaScript variable, {noArrow}, is available to turn off submenu hinting.

- Borders: boxBorder, boxBorderColor, intBorder, and backColor. See Table 1 on page 7 for the correct values to match standards.
 - Backgrounds: bgOnColor and bgOffColor. See Table 1 on page 7 for the correct values to match standards.
2. For each item in the top menu, add a menuItem element inside menuSystem. Enter the label for the menu item inside the corresponding menuItem element. An example of two top-level menu items is shown below:

```
var theXMLIs= theXMLIs +
  '<menuSystem useInternalTriggers="yes" layerTriggerPrefix="zzz">'+
  '<menuItem>Products</menuItem>'+
  '<menuItem>About Our Company</menuItem>'+
  '</menuSystem>';
```

Most of the attributes set at menuSystem default to the individual menu items. But you can also control the look of individual menu items. See Table 3 on page 31 for a list of the attributes that you can use.

Note: One attribute that only applies to top-level menu items is localWidth which allows you to set specific widths to display that label but have any submenu items use a standard width.

3. For each option that should be available from the top-level menu items, you must nest a `menuItem` element inside the appropriate top-level `menuItem`. Add an `href` attribute with the URL of the page that should open for that menu item (or the JavaScript function that should be executed).

Important: the XML Menuing System cannot currently support some specific characters within URLs in the `href` attribute. See the Known Bugs, V1.5, V1.6, and V1.7 section on page 46 for specific information.

In addition, you may need to escape certain characters within JavaScript function calls, such as single quotes around string arguments. Because the call to the function is itself within a JavaScript string, these characters should be escaped with `\'`.

An example of options off the 'Product' menu from the previous example is shown below:

```
var theXMLIs= theXMLIs +
'<menuSystem useInternalTriggers="yes" layerTriggerPrefix="zzz">'+
'<menuItem>Products'+
'<menuItem href="products/turbines.html">Turbines</menuItem>'+
'<menuItem href="products/parts.html">Parts</menuItem>'+
'<menuItem href="products/installs.html">Installation</menuItem>'+
'</menuItem>'+
'<menuItem>About Our Company</menuItem>'+
'</menuSystem>';
```

Continue adding `menuItem` elements to define the full menu tree. You should **not** define more than three levels of menus. If the application uses frames, only one level of menu items is allowed.

Technical Syntax Issues

Keep the following syntax issues in mind while coding configuration information:

- The variable name **theXMLIs** is required and cannot be changed.
- Unless you put all of your XML information on one line, you need to follow normal JavaScript syntax for defining multi-line strings. For example:

Example 3. Multi-Line JavaScript Variable for Menus

```
var theXMLIs= theXMLIs+
    ' ----- xml configuration information -----'+
    ' ----- xml information continued -----'+
    ' ----- xml information continued -----'+
    ' ----- xml information continued -----';
```

- Any use of JavaScript functions within the href attribute may need to use escape characters because the function call is itself within a JavaScript variable. For example:

```
'<menuItem href="javascript:doOrder('\add\',">Add to Order</menuItem
```

This JavaScript function takes one string parameter. The single quote marks that delimit the string must be escaped.

- The XML parser that parses configuration information is designed for a simple, specific XML structure. You may use escape characters within the structure.

Although the configuration uses well-formed XML, there are some limitations to the content if you use an XML editing tool to produce the configuration information. The following types of XML markup **cannot** be in configuration information:

- The XML declaration, such as `<?xml version="1.0" encoding="utf-8" ?>`
- A DOCTYPE declaration, **with or without** an internal subset.
- Comments, such as `<!-- menu level 1 -->`
- Processing instructions, such as `<?mySys myattr="1" ?>`
- Any references to entities **not including** any Unicode character references.
- XHTML elements in the content, including elements such as `img` for inserting images.

Using Externally Triggered Menus

Externally triggered menus use XML configuration information to define the number of top-level menu items and what their children are (second and third levels). The visible top-level menu items, however, are displayed on the page using elements in the page.

The menuing system suppresses the configuration information for the top-level menuitem elements. The page elements that do display the top-level menu items must also be coded to trigger the display of second-level menus when users hover over them.

To accomplish this, you must configure the menus for external triggers and then code the trigger page elements.

Configuring the Menus for External Triggers

In the example shown below, the menuitem elements named 'Menu item 1', 'Menu item 2', and 'Menu item 3' are not rendered by the menuing system engine. However, elements in the page act as triggers so that the second level (drop down) menu items are rendered properly in response to user mouse actions.

Example 4. Menu Configuration for External Triggers

```
var theXMLIs= theXMLIs +
'<menuSystem layerTriggerPrefix="extrn" useInternalTriggers="no" '+
'backgroundColor="#ffffff" bgOnColor="#C1DDBF" bgOffColor="#aed0ab" '+
'fontSize="12px" fontColor="#000000" fontOnColor="#006600" >'+
'  <menuItem startX="50" startY="80" width="85" height="20">Menu item 1'+
'    <menuItem href="menu11.jsp">Menu item 1.1</menuItem>'+
'    <menuItem href="menu12.jsp">Menu item 1.2</menuItem>'+
'  </menuItem>'+
'  <menuItem startX="135" startY="80" width="85" height="20">Menu item 2'+
'    <menuItem href="menu21.jsp">Menu item 2.1</menuItem>'+
'    <menuItem href="menu22.jsp">Menu item 2.2</menuItem>'+
'  </menuItem>'+
'  <menuItem startX="220" startY="80" width="85" height="20">Menu item 3'+
'    <menuItem href="menu31.jsp">Menu item 3.1</menuItem>'+
'    <menuItem href="menu32.jsp">Menu item 3.2</menuItem>'+
'  </menuItem>'+
'</menuSystem>';
```

The primary differences in menu configuration information for externally triggered menus are:

- The `useInternalTriggers` attribute is set to "no".
- Positioning and size attributes should be set on the first-level menuitem elements. This includes the `startX`, `startY`, `eTOffsetX`, `eTOffsetY`, `width`, and `height` attributes.

To handle this correctly, the top-left corner where each second level menu will be displayed is determined from the `startX` value (horizontal) and the `startY + height` value (vertical). Coordinate this with the actual page elements that are being used as triggers to get a smooth visual effect when users hover over the top level menu items in the page. Use the `eTOffsetX` and `eTOffsetY` attributes to fine tune the synchronization.

Defining the External Menu Triggers in the Page

To make the connection between trigger elements on the page and the correct menu configuration information, the trigger elements must be coded according to the following rules:

- The trigger element added to a page for a top-level menu item must be a child of an anchor (a) element.
- The parent a element must contain the following attributes:
 - `href` attribute with a value of "javascript:doNothing()."
 - `onMouseOver` attribute that triggers the menuing system's `handleOver()` function. The parameter passed in `handleOver` is a code identifying which top-level menu item this external trigger should emulate.
 - `onMouseOut` attribute that triggers the menuing system's `handleOut()` function. The parameter passed in `handleOut` is a code identifying which top-level menu item this external trigger should emulate.

These functions trigger the menu engine to render the second level of menu items and to respond to user mouse actions.

- The parent a element must be nested within (a child or descendant of) a div element.
- The parent div element must have an `id` attribute with a value that conforms to the naming convention: *menuPrefixmmenu-item-number-to-be-triggeredTrigger*.

An example of the code within the page that acts as the external trigger for Example 4 on page 21 is shown below. Note that the number in the `id` value for each trigger is **zero-based** -- zero works for the first menu item, one for the second menu item, and so on.

Example 5. External Menu Trigger Code in the Page

```
<div id="extrnm0Trigger">
```


Example 5. External Menu Trigger Code in the Page (continued)

```
<a href="javascript:doNothing()"
  onMouseOver="handleOver('extrnm0')"
  onMouseOut="handleOut('extrnm0')">Trigger for Menu Item 1</a>
</div>
<div id="extrnm1Trigger">
  <a href="javascript:doNothing()"
    onMouseOver="handleOver('extrnm1')"
    onMouseOut="handleOut('extrnm1')">Trigger for Menu Item 2</a>
</div>
<div id="extrnm2Trigger">
  <a href="javascript:doNothing()"
    onMouseOver="handleOver('extrnm2')"
    onMouseOut="handleOut('extrnm2')">Trigger for Menu Item 3</a>
</div>
```

Using Multiple MenuSystem Elements

You may define multiple menuSystem elements within the menu configuration file for Intranet web sites or applications to provide local menus (specific to one page or a set of pages). Intranet web sites and applications that use menus **must consistently** choose either horizontal or vertical menus for their site-wide menus. Local menus may be added and may use the opposite orientation.

The following example shows a horizontal menu with different widths for each top level menu item. Each item is configured as one menuSystem element. Note that the layerTriggerPrefix attribute is different for **each** menuSystem element defined.

Example 6. Multiple Menu Systems

```
var theXMLIs= theXMLIs +
'<menuSystem startX="49" startY="'+ navMenuY +' " width="89" height="18" '+
'layerTriggerPrefix="aaa" backgroundColor="#ffffff" useInternalTriggers="yes" '+
'fontFamily="Arial" fontSize="12px" fontColor="#000000" '+
'fontOnColor="#990000" bgOnColor="#999966" bgOffColor="#dbdac5">'+
'<menuItem width="180">News'+
'<menuItem href="ourNews.jsp">Our News</menuItem>'+
'<menuItem href="industryNews.jsp">Industry News</menuItem>'+
'</menuItem>'+
'</menuSystem>'+

'<menuSystem startX="138" startY="'+ navMenuY +' " width="58" height="18" '+
'layerTriggerPrefix="bbb" backgroundColor="#ffffff" useInternalTriggers="yes" '+
'fontFamily="Arial" fontSize="12px" fontColor="#000000" '+
'fontOnColor="#990000" bgOnColor="#999966" bgOffColor="#dbdac5">'+
'<menuItem width="180" href="products.jsp">Products'+
'</menuItem>'+
'</menuSystem>'+

'<menuSystem startX="196" startY="'+ navMenuY +' " width="138" height="18" '+
'layerTriggerPrefix="ccc" backgroundColor="#ffffff" useInternalTriggers="yes" '+
'fontFamily="Arial" fontSize="12px" fontColor="#000000" '+
'fontOnColor="#990000" bgOnColor="#999966" bgOffColor="#dbdac5">'+
'<menuItem width="180">Customer Service'+
'<menuItem href="aboutUs.jsp">About Us</menuItem>'+
'<menuItem href="support.jsp">Support</menuItem>'+
'<menuItem href="feedback.jsp">Feedback</menuItem>'+
'</menuItem>'+
'</menuSystem>';
```

Example 6. Multiple Menu Systems (continued)

Assigning Menus Dynamically

In some applications or web sites, the options that users should be able to see and access may vary depending on information about the user, the current session, or the current state. To handle this requirement, use the {menuFileURL} JSP variable to define which XML configuration information should be rendered.

Each potential variation of the menus is set up in a different menu configuration file. The actual file that the XML Menuing System uses to render the menus can then be determined using JSP logic to assign a file to the {menuFileURL} variable within each page.

The following example illustrates JSP code that selects one of four menu configuration files based on whether the user is logged in, and if so, the user's access level. Users who have not logged in see the default menu, from menuConfig.js. Users who have logged in see menus from menuConfigSilver.js, menuConfigGold.js, or menuConfigBlue.js based on their access level value.

Example 7. JSP Logic for Dynamic Menus

```
<%@ include file="applicationSettings.jspi" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>JSP Template</title>
<%
    menuFileURL="menuConfig";
    if (appuser.isLoggedIn())
        { menuFileURL += "Blue";
        } else if (accesslevel == 'silver')
        { menuFileURL += "Silver";
        } else if (accesslevel == 'gold')
        { menuFileURL += "Gold";
        }
    menuFileURL += ".js";
%>
<%@ include file="/global/extranet/1.2intranet/1.0/jsp/headConfig.jspi"%>
```

These example menus provide access to three levels of users as shown in Table 2 on page 28.

Table 2. Sample Menu Function Access by User Level

Menu and Function	User Level Blue	User Level Silver	User Level Gold
View Menu: View Requests item	yes	yes	yes
View Menu: View Confirmations item	yes	yes	yes
Edit Menu: Create Requests item	yes	yes	yes
Edit Menu: Edit/Submit Requests item		yes	yes
Delete Menu: Delete a Request			yes
Delete Menu: Delete Old Requests			yes

Simplified examples of the three menu files are shown below.

Example 8. Sample Blue Menu Configuration

```

var theXMLIs= theXMLIs +
'<menuSystem useInternalTrigger="yes" layerTiggerPrefix="abc">'+
  '<menuItem>View Menu'+
    '<menuItem href="viewreq.jsp">View Requests</menuItem>'+
    '<menuItem href="viewconf.jsp">View Confirmations</menuItem>'+
  '</menuItem>'+
  '<menuItem>Edit Menu'+
    '<menuItem href="newreq.jsp">Create Requests</menuItem>'+
  '</menuItem>'+
  '<menuItem> '+
  '</menuItem>'+
'</menuSystem>';

```

Note that menuitem elements are defined for all three menus, but menus where a blue user should have no access (the third one) have no label and no children menuitem elements.

Example 9. Sample Silver Menu Configuration

```

var theXMLIs= theXMLIs +
'<menuSystem useInternalTrigger="yes" layerTriggerPrefix="abc">'+
  '<menuItem>View Menu'+
    '<menuItem href="viewreq.jsp">View Requests</menuItem>'+

```

Example 9. Sample Silver Menu Configuration (continued)

```
'<menuItem href="viewconf.jsp">View Confirmations</menuItem>'+  
'</menuItem>'+  
'<menuItem>Edit Menu'+  
  '<menuItem href="newreq.jsp">Create Requests</menuItem>'+  
  '<menuItem href="sendreq.jsp">Edit/Submit Requests</menuItem>'+  
'</menuItem>'+  
'<menuItem>'+  
'</menuItem>'+  
'</menuSystem>';
```

With the silver version of the menu, the third first-level menuitem is still empty. But an additional child item has been added to the Edit Menu for the additional function that silver users may access.

With the gold version of the menu, all three first-level menuitem elements have labels and children items, allowing access to the full menu tree.

Example 10. Sample Gold Menu Configuration

```
var theXMLIs= theXMLIs +  
'<menuSystem useInternalTrigger="yes" layerTriggerPrefix="abc">'+  
  '<menuItem>View Menu'+  
    '<menuItem href="viewreq.jsp">View Requests</menuItem>'+  
    '<menuItem href="viewconf.jsp">View Confirmations</menuItem>'+  
  '</menuItem>'+  
  '<menuItem>Edit Menu'+  
    '<menuItem href="newreq.jsp">Create Requests</menuItem>'+  
    '<menuItem href="sendreq.jsp">Edit/Submit Requests</menuItem>'+  
  '</menuItem>'+  
  '<menuItem>Delete Requests'+  
    '<menuItem href="dellreq.jsp">Delete a Request</menuItem>'+  
    '<menuItem href="deloldreq.jsp">Delete Old Requests</menuItem>'+  
  '</menuItem>'+  
'</menuSystem>';
```

XML Menu Configuration Markup

This section contains detailed reference information on the XML elements and attributes that define menu configuration information.

Elements

There are only two elements within XML menu configuration information:

- **menuSystem**: this is the root element for configuration information. Use it to define the scope of one menu system and to set menu-wide configuration parameters.
- **menuItem**: use this element to define the structure of the menu system, the labels for each menu item and to define any item-specific configuration information. You may nest menuItem elements to define the hierarchy of the menu.

Note: You should **not** create a menu system with more than three levels and limiting it to two levels is strongly advised. Instead, create multiple menu systems.

The attributes that define the menu configuration appear on the children nodes when applied to the root menuSystem.

Table 3. Valid Attributes for menuSystem and menuItem

Attributes for menuSystem	Attributes for menuItem
arrowHref	arrowHref
arrowPadding	arrowPadding
backColor	backColor
bgOnColor	bgOnColor
bgOffColor	bgOffColor
boxBorder	boxBorder
boxBorderColor	boxBorderColor
drop	fontColor
eTOffsetX	fontOnColor
eTOffsetY	fontSize
fontColor	fontWeight
fontFamily	height

Table 3. Valid Attributes for menuSystem and menuItem (continued)

Attributes for menuSystem	Attributes for menuItem
fontOnColor	href
fontSize	intBorder
fontWeight	localWidth
height	width
hideSelect	
intBorder	
layerTriggerPrefix	
orientation	
startX	
startY	
useInternalTriggers	
width	

Attributes: A-E

arrowHref

A graphic to appear after the label of any submenus that have children. By default, all menu items with children (and thus have a submenu) display a small graphic arrow (a triangle) that points to the right and is right aligned within the width allowed. Use this attribute, along with the arrowPadding attribute to control what graphic displays (or if a graphic should display) and the placement of these submenu hints.

Use this attribute to specify another graphic to use as a submenu hint. The size of the graphic should be set to fit within both the height and width of the menu item and the background should be transparent.

To **turn off** submenu hinting, you must set this attribute to the standard JavaScript variable {noArrow}. This variable points to a single pixel, transparent graphic.

Valid entry:

Any string that is a valid URL to a graphic image. This URL generally must be relative to the web application root directory.

This attribute may be set with one of two variables to use the standard graphic for horizontal menus. Use `arrowHref="'+topLevelArrow+'"` for top level menu items and `arrowHref="'+subLevelArrow+'"` for second level menu items.

Use `arrowHref="'+noArrow+'"`, the final JavaScript variable, to **turn off** submenu hinting.

Required:

No

Default value:

"img/arrowredDown.gif"

arrowPadding

Controls the placement of graphics used for submenu hinting. By default, these graphics are right aligned in the horizontal space available. If this attribute is set, the graphic appears directly after the text of the menu item with the space specified between the text and the graphic.

Like `arrowHref`, this attribute should be set on the parent element to affect submenu hinting for all children `menulitem`. Put simply: 1) set this attribute on `menuSystem` to affect first-level menu items, 2) set it on first-level `menulitem` elements to affect second-level menu items, and so on.

Important: To insure compatability for Netscape, you **must** also define the {NSarrowPad} variable in your menu configuration file(s), such as menuConfig.js. This variable should define the arrowPadding attribute with the same width. Then use this variable on the menuItem children where padding should be effective. For example:

```
var NSarrowPadding='arrowPadding="10" ';
var theXMLIs = theXMLIs +
'<menuSystem ... arrowHref="'+topLevelArrow+' '
  arrowPadding="10" ...>'+
  '<menuItem
    arrowHref="'+subLevelArrow+NSarrowPadding+'>
    Top Level Menu 1'+
    '<menuItem href="viewreq.jsp">
      Sub Menu 1.1</menuItem>'+
    '<menuItem href="viewconf.jsp">
      Sub Menu 1.2</menuItem>'+
  '</menuItem>'+
  '<menuItem
    arrowHref="'+subLevelArrow+NSarrowPadding+'>
    Top Level Menu 2'+
    '<menuItem href="newreq.jsp">
      Sub Menu 2.1</menuItem>'+
    '<menuItem href="sendreq.jsp">
      Sub Menu 2.1</menuItem>'+
  '</menuItem>'+
  ...
```

Valid entry:

Any positive integer (padding measurement is in pixels). Set this to 10 for top level menu items in horizontal menus **only**.

Required:

No

Default value:

null

backColor

The color of the background that shows as an interior border (inter-item) or as padding. See also the intBorder on page 41 attribute.

The default value is white, which creates white space rather than a visible border if the background of your page is white.

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900'. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#ffffff

bgOffColor

The color of the background for menu items when a menu item is 'off' (not being moused over).

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900'. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#f5f5ea

bgOnColor

The color of the background for menu items when a menu item is 'on' (being moused over).

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900'. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#888871

boxBorder

Defines the width in pixels of a consistent border around the entire list of submenu items (children menuitem elements). This does not define inter-item borders (see intBorder) and does not affect horizontal menu items. See Figure 1 on page 11 for an illustration.

Valid entry:

Any positive integer. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on standard sizing for menus implemented in the GEPS Global UI.

Required:

No

Default value:

1

boxBorderColor

Defines the color used for a border around the entire list of submenu items (see boxBorder). This does not define inter-item borders (see intBorder) and does not affect horizontal menu items.

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900.' See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#000000

Attributes: D-E

drop

Sets the direction that items on this menu drop from their parent -- up or down. This attribute is only valid on menuSystem.

Note: For vertical menus, this value may be ignored if the size of the submenu exceeds the space within the window. For example, if the menu is set to drop down but there isn't enough space to display all the menu items, the menu automatically drops up instead.

Valid entry:

String matching 'down' or 'up'

Required:

No

Default value:

Down

eTOffsetX

Defines the horizontal offset in pixels, relative to an externally triggered menu item, to shift the display of the submenu (children menuitem elements). Positive values shift the submenu right; negative values shift the submenu left.

This attribute is only valid on menuSystem.

Valid entry:

Any integer

Required:

No

Default value:

175

eTOffsetY

Defines the vertical offset in pixels, relative to an externally triggered menu item, to shift the display of the submenu (children menuitem elements). Positive values shift the submenu down; negative values shift the submenu up.

This attribute is only valid on menuSystem.

Valid entry:

Any integer.

Required:

No

Default value:

20

Attributes: F-M

fontColor

The color of the item labels when the menu item is 'off' (not moused over).

Note: In Netscape 4.x, this attribute has no effect when it is assigned to individual menu items. See the Known Bugs, V1.5, V1.6, and V1.7 section on page 46 for more information.

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900'. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#443322

fontFamily

The name of the font to use for item labels in the menu system. This attribute is only valid on menuSystem.

Valid entry:

Any string matching a supported font name

Required:

No

Default value:

Verdana

fontOnColor

The color of the item labels when a menu item is 'on.'

Note: In Netscape 4.x, this attribute has no effect when it is assigned to individual menu items. See the Known Bugs, V1.5, V1.6, and V1.7 section on page 46 for more information.

Valid entry:

Any string that is a valid hexadecimal color value, such as '#339900'. See the Menu Coding section on page 13 for information on valid colors for menus implemented in the GEPS Global UI.

Required:

No

Default value:

#ffffff

fontSize

The size, in points, to use for item labels of this menu system or menu item.

Valid entry:

Any positive integer excluding zero that is a valid font size. The current unit is points. Future releases will use pixels as the unit of measurement.

See theMenu Coding section on page 13 for information on valid sizes for menus implemented in the GEPS Global UI.

Required:

No

Default value:

12px

fontWeight

The weight of the font (bold or normal).

Valid entry:

A string matching 'bold' or 'normal'

Required:

No

Default value:

Normal

globalShiftX

Sets the horizontal overlapping of children to their parent menuitem. A negative number shifts the menuitem left (towards the parent) and a positive number shifts them right (away from the parent).

Valid entry:

Any integer including zero

Required:

No

Default value:

-4 (interger)

globalShiftY

Sets the vertical overlapping of children to their parent menuitem. A negative number shifts the menuitem up (towards the parent) and a positive number shifts them down (away from the parent).

Valid entry:

Any integer including zero

Required:

No

Default value:

2 (integer)

Attributes: H-M**height**

The visible height of each menu item that is a child of the menu system or menu item. Best set on menuSystem to simplify the configuration and to present a consistent and balanced display.

Note: Padding values must fit within the height value. Borders, both overall and inter-item, are not affected.

Valid entry:

Any positive integer excluding zero.

Required:

No

Default value:

20 pixels

hideSelect

Controls the display of Select boxes when pages with menus first load in Internet Explorer. IE has a bug regarding 'infinite z-indexes' which can cause Select boxes to always display on top, even if the user mouses over a menu item that should be the current top level.

This attribute actually controls whether IE creates an array of pointers to the Select boxes within a page. This array can then be used to set the style.display property to "none" for the select boxes when a menu system is visible. This attribute is only valid on menuSystem.

Valid entry:

A string of "yes" or "no".

Required:

No

Default value:

"no"

href

The URL or JavaScript function to activate for this menu item. This attribute is only available on menuItem. The content can be any valid URL that does not contain a forbidden character. JavaScript commands are allowed. See also the Known Bugs, V1.5, V1.6, and V1.7 section on page 46 for information on characters that are must be escaped.

Valid entry:

Any string that is a valid URL or JavaScript function call

Required:

No

Default value:

Null

intBorder

Defines the width of a consistent 'interior' border for children menu items of the menuItem or menuSystem element it is used on. This border uses the color defined in the backColor attribute and thus defines a grid around submenu items. See the Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for the standard values to use for menus.

This inter-item border displays **in addition** to the width of border defined in boxBorder attribute. See Figure 1 on page 11 for an illustration of the various area relationships.

Valid entry:

Zero or any positive integer.

Required:

No

Default value:

0

layerTriggerPrefix

A unique string identifier that is added to the beginning of all ID values for layers associated with this menu system. This identifier differentiates layers for this menu from generic layers within the page and from layers for other menus. This attribute is only valid on menuSystem.

Valid entry:

Any string of characters that could be used as a valid variable name in JavaScript

Required:

Yes

Default value:

Null

localWidth

The width, in pixels, to use to display the label of a **top-level, horizontal** menu item. This allows you to position top-level menu items in horizontal menus relative to the size of their label. The width attribute still controls the width of all children menu items.

Valid entry:

Any positive integer excluding zero.

Required:

No

Default value:

null

Attributes: N-Z

orientation

Controls the horizontal or vertical orientation for top level menu items. If this attribute is omitted or set to 'h', the top level menu items display horizontally. If this is set to 'v', the top level menu items display vertically. This attribute is only valid on menuSystem.

Valid entry:

A string matching 'h' (horizontal) or 'v' (vertical)

Required:

No

Default value:

'h' (for horizontal menus)

startX

The X coordinate for the top left corner where the menu should display. This attribute is only valid on menuSystem. See also the eTOffsetX attribute for positioning with externally triggered menus.

Valid entry:

Any integer greater than or equal to zero

Required:

No

Default value:

0

startY

The Y coordinate for the top left corner where this menu system should display. This attribute is only valid on menuSystem. See also the eTOffsetY attribute for positioning with externally triggered menus.

Valid entry:

Any integer greater than or equal to zero. This attribute may be set as `startY="'+navMenuY+'"` to snug the menu directly under the standard masthead.

Required:

No

Default value:

0

useInternalTriggers

Controls the visibility of the top-level menu items and determines whether menus are 'internally triggered' or 'externally triggered.' This attribute is only valid on menuSystem.

A value of 'yes' defines internally triggered menus and causes the menu engine to render the top-level menu items based on XML configuration information.

A value of 'no' defines externally triggered menus and stops the menu engine from rendering the top-level menu items. Instead, these menu items must be coded as external triggers within every page where the menu should appear. This allows greater control of the visible appearance of the top level menus, but requires additional work, especially if menus are dynamic. See the Using Externally Triggered Menus section on page 21 for an example and coding rules to use for external triggers.

Valid entry:

A string matching 'yes' or 'no'

Required:

Yes

Default value:

Null

width

The visible width in pixels of each menu item that is a child of the menu system or menu item. Best set on menuSystem to simplify the configuration and present a consistent and balanced display.

Valid entry:

Any positive integer excluding zero

Required:

No

Default value:

150 pixels

Appendix A: Release Notes

Version 1.7b of the XML Menu System is deployed in Version 1.0 of the Global UI. All release notes refer to the XML Menu System version, rather than the deployed version of the Global UI.

Version 1.5, 1.6, and 1.7

New and Changed

- The mouse cursor now changes to a "hand" (or to the user's configured icon for "link select" mode) when the mouse hovers over a menu item. Previously, the default "arrow" pointer (or configured icon) was used.
- Parent menu items remain highlighted (with the "on" color choices) when the mouse is over any of their children menu items.
- Several attributes have been added:
 - localWidth: use this to change the width of top-level menu items. The width attribute still controls the width of all children items.
 - arrowHref and arrowPadding: use this to include small graphics after the label of a menu item and control its placement and alignment. See Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on the specific standards to follow for submenu hinting.
 - hideSelect: use this to correct Internet Explorer bugs in displaying menus and Select boxes on the same page.
 - eTOffsetX and eTOffsetY: use these to control the offsets of children menu items in externally triggered menus.
 - boxBorder, boxBorderColor, and intBorder: use these to define borders for menu items. See Colors, Fonts, Borders, Roll-overs, and Other Effects section on page 6 for information on the specific standards to follow for menu borders.
 - orientation: use this to control the horizontal or vertical display of top-level menu items.
- Support for Netscape 4.7 has been removed in version 1.7b and the rendering algorithm simplified. As a result, this version must be used with version 1.1 of xmlParser_lib.js.

Note: Netscape 4.7 support is still supplied by version 1.6d of the XML Menuing System. The Global UI automatically switches to the correct version of the XML Menuing System engine for Intranet web site and applications to ensure support for users on Netscape 4.7.

- Several attributes are no longer relevant using the new rendering engine. These have been removed:
 - delay
 - Nx4CssFix
 - All the padding attributes (padding, paddingBottom, paddingLeft, paddingRight, paddingTop)
 - Some of the implementation details (already present in the Global UI) have been simplified.

Known Bugs, V1.5, V1.6, and V1.7

- Four bugs from version 1.3 remain open (the Netscape 4.7 bugs only apply to the Intranet):
 - Netscape 4.x fontOnColor is not implemented.
 - Netscape 4.x does not support the fontColor and fontOnColor properties at the menuItem level.
 - Netscape 5(6) issues with DHTML in XHTML remains open.
 - Netscape 4.x initial page error remains open.

Version 1.4

New and Changed

- Two new attributes have been added, globalShiftX and globalShiftY, to control the overlap and spacing between children menuItem elements and their parent. Use these attributes to create shifts or overlaps to make the relationship between the active menu display more apparent.
- The display of children menu items is now sensitive to the right edge of the browser window (previously, the next menu level always displayed to the right which sometimes meant the menu was not visible). If displaying a subsidiary menu to the right forces the menu off the right edge of the browser window, the subsidiary menu now display to the left of the menu item automatically.

Subsidiary menus off a menu that was forced to display to the left still display to the right as long as the menu display still fits within the browser window. This can cause a 'zip-zag' pattern of menu display for menus that are close to the right browser window edge.
- Fixed a bug with the z-index for the back layer of child menuItem being below the z-index of the parent menuItem.

Known Bugs, V1.4

All known bugs from version 1.3 remain open, except the following method to handle characters within the href attribute that are not properly interpreted.

You may substitute the Unicode character entity (either decimal or hexadecimal) for any characters that the href attribute does not properly

interpret. The characters and their decimal Unicode entity that have specifically been identified as problematic are:

- " (Unicode ")
- = (Unicode =)
- \ (Unicode \)

Version 1.3

Known Bugs

- The href attribute does not properly interpret the following characters within a URL: =, \, and the double quotation mark (").
- The fontColor and fontOnColor attributes are ignored on individual menu items in Netscape 4.x. Only font colors specified at the menuSystem element are used.

Netscape 4.x does not expose the font color attribute properly in its implementation of the DOM. As a result, making fontColor and fontOnColor work properly would require rewriting the HTML markup within the menuitem layer and reassigning its stylesheet values every time a change of color was desired. The fix to this issue significantly degraded the speed of the menu system.

- Netscape 5(6) rendering issues occur with DHTML in XHTML. The temporary fix for this bug is to code pages following XHTML syntax, but use an HTML 4.0 DOCTYPE declaration and omit the XML declaration.

Netscape 5(6) has issues with DHTML in general when the DOCTYPE is declared as XHTML. Initial research into the problem revealed that it may be due to the two rendering modes that Netscape 5 uses: "quirks" and "standards." More research is required to confirm this and to determine which mode is being used in an XHTML document. For more information on this issue, see www.w3c.org/TR/xhtml1/.

- The very first page loaded after Netscape 4.x is initialized may crash if it uses the DHTML functions in the standardLib.js JavaScript library. Refreshing the page corrects the problem.

Netscape 4.x produces an error on some pages using DHTML functions from the standard JavaScript library. If it is the first page to be displayed when Netscape first loads, then the page crashes. Unfortunately, the problem is so severe that even simple JavaScript solutions, such as making a rudimentary change to the window.location object to automatically refresh the page if there are no entries in the window.history array, cannot be implemented as a workaround.

The menuing systems is designed to work in Microsoft Internet Explorer 4.0 and higher and Netscape Communicator 4.7 (Communicator 4.7 or Navigator 4.08) and higher.

Appendix B: Technical Implementation

The XML Menu System consists of two JavaScript libraries, xmlMenu_lib.js and xmlParser_lib.js, and as many configuration files as needed to implement your menus. This system also depends on the GEPS Standard JavaScript Library, standard_lib.js. To implement the XML Menu System separately from the GEPS Global UI, you must have all three libraries.

This appendix outlines the requirements to use the XML Menuing System separately from the GEPS Global UI. A full example of the required markup is shown below.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>Example Page with XML menuing system</title>
  <script language="javascript1.2" type="text/javascript" src="standard_lib.js">
  <script language="javascript1.2" type="text/javascript" src="menuConfig.js"></
  <script language="javascript1.2" type="text/javascript" src="xmlParser_lib.js"
  <script language="javascript1.2" type="text/javascript" src="xmlMenu_lib.js"><
</head>
<body>

  Put your page content here

</body>
</html>
```

Complete the following steps:

1. **Include the JavaScript Library files:** in each page where a menu should be implemented, add four script elements in the head area of the page.

Four files are required: standard_lib.js, menuConfig.js, xmlParser_lib.js, and xmlMenu_lib.js. An example of the head element with the required script elements is shown below:

Example 11. Menu System JavaScript Files

```
<head>
<title>Page Title</title>
<script language="javascript1.2" type="text/javascript"
  src="standard_lib.js"></script>
<script language="javascript1.2" type="text/javascript"
  src="menuConfig.js"></script>
<script language="javascript1.2" type="text/javascript"
  src="xmlParser_lib.js"></script>
<script language="javascript1.2" type="text/javascript"
  src="xmlMenu_lib.js"></script>
</head>
```

Caution: The script elements must be listed in the page in this **exact** order. Changes to the order will cause errors.

2. **Configure the Menu Content in menuConfig.js:** the actual labels and actions for the menu are defined in a JavaScript string that contains XML markup. This information resides in menuConfig.js.

To complete this step, you must be familiar with XML syntax. Create a JavaScript file with this name and declare a variable named "**theXMLs**" within this file.

Note: This exact variable name is required and is case-sensitive as it is used in methods within the JavaScript libraries that support the XML menuing system.

Then complete the XML configuration information for your menus. See the Configuring Basic Menus section on page 17 for more details.

Glossary

ancestor

[XML] An XML element containing another element--its descendant--anywhere within it. For example: In the XML structure `<book><chapter><para></para></chapter></book>` the `para` element is the descendant of both the `chapter` and `book` elements. `Book` and `chapter` are ancestors of `para`.

The ancestor element that is directly above a descendant is often referred to as the parent. The descendant element that is directly below an ancestor is referred to as the child. Thus, in the example above, `book` is the parent of `chapter` and `chapter` is the child of `book`.

application

[UIStd] For user interface standards, an application is a set of web pages (a site) that present an organized set of functionality to users generally related to one specific business domain or one specific set of user tasks. Thus applications are directly involved in completing transactions and tasks rather than simply in providing information.

Applications are always dynamic (not just simple static HTML pages). Because, web sites may also be dynamic, a site is an application if any of the following rules is true:

- Processing for the site uses or updates data in a database that exists for business purposes other than solely to provide content to the site. This typically also means that the database stores data involved directly in business transactions and may also be used by other applications or systems.
- Processing for the site integrates (receives or sends data or otherwise directly interacts) with other applications or systems:
 - Within GEPS
 - Belonging to customers or vendors

Simply sending an email, for example, does not make a site an application. But sending an order directly to a vendor system or receiving documents for a work-flow from a customer system does constitute 'integration' and thus makes a site an application.

attribute

[XML] Generally, an XML attribute is a parameter or setting on an element that provides additional information about

the element. Attributes are stored in the start tag of an element as name="value" pairs or are assigned default values in attribute declarations within the DTD.

In documents attributes represent characteristics about the data. In messages and configuration files, attributes are the data.

child element

[XML] An element nested directly inside another element. In `<msg><msgelement><elementptr/></msgelement></msg>` the `msgelement` element is the child element of the `msg` element, and the `elementptr` element is the child of the `msgelement` element. The `elementptr` element not the child of the `msg` element. It is a descendant.

DTD

[XML] *Document Type Definition:* A file, written for SGML or XML, that defines a specific type of document, message, or configuration. The DTD identifies the elements that can be used and specifies their valid uses, including where they can occur and how they can be nested. The DTD defines a content model for each element that contains other elements or attributes.

DOM

Document Object Model: The specification for how objects in a Web page are displayed. The DOM defines the attributes that are associated with each object, and specifies how the objects and attributes can be used. DHTML uses the DOM to dynamically change the appearance of Web pages once they are downloaded to a user's browser.

[XML] In XML, the DOM is the structure of the document as defined in the DTD.

element

[XML] The unit forming the basic structure of XML documents. Elements may contain attributes (in their start tags), other elements, and textual content.

[XML] *empty element:* An XML element without any content. Empty elements may contain only attributes. Typically used to insert link, images, or objects.

[JSP] A portion of a JSP page that is recognized by the JSP translator. A JSP element can be a directive, an action, or a script.

end tag

[XML] In XML, a tag that marks the end of an element. An end tag uses the syntax `</Name>` where `Name` is the same as the element name that was used in the start tag for the element.

parent element

[XML] An element directly containing another element. In `<msg><msgelement><elementptr/></msgelement></msg>` the `msg` element is the parent element of the `msgelement`

element, and the msgelement element is the parent of the elementptr element, but the msg element is not the parent of the elementptr element.

start tag

[XML] In XML, SGML, or HTML, the opening tag that identifies the beginning of an element within a document and may contain attribute values. Start tags typically use the syntax `<elementNamed attribute="value">` but may use the syntax `<elementNamed attribute="value"/>` if the element has no textual content or sub-elements.

tag

In a markup language, such as XML or HTML, a tag is a descriptor and delimiter for an element. If the element contains content, both a start and end tag are used to surround the content. If the element does not contain content, it is marked by a single empty tag.

Unicode

A standard (Unicode Worldwide Character Standard) for the interchange, processing, and display of different languages. Currently, the Unicode standard contains 34,168 coded characters from 24 supported language scripts covering the principal written languages of the world. The Unicode standard is kept in sync with ISO/IEC 10464, which uniquely identifies the characters and glyphs from the most modern and from many ancient languages.

web site

[UIStd] For user interface standards, the simplest definition of a web site is a site (a set of web pages) that is **not** an application. Web sites are primarily built to present information. Web sites may consist of solely static content or dynamic content or both. Web sites may also include access to applications, but are generally not involved in generating transactions and completing tasks.

XML

Extensible Markup Language: An international standard markup specification used to design ways of describing the structure and content of information. XML was defined by the W3C to provide a strict set of standards for document syntax while allowing developers, organizations, and communities to define their own vocabularies for use on the Internet.

XSL

Extensible Style Language: A style sheet standard developed and maintained by the W3C. XSL uses template rules (written using XML) to transform documents into 'formatting objects', which are then presented on screen, in print, or in other media.

XSLT

XSL Transformations: An extension of the XSL standard used to transform the data structure of an XML file into a different structure, such as HTML, or another XML file with

a different structure. XSLT code is referred to as a stylesheet and can be combined with an XSL stylesheet to define the appearance of the new file, or used independently.

Index

A

Additional Resources, 3
arrowHref menu attribute, 32
arrowPadding menu attribute, 33
Assigning Menus Dynamically, 27
attributes, 32, 42
 See also menu attributes
Attributes: A-E, 32
Attributes: D-E, 36
Attributes: F-M, 37
Attributes: H-M, 40
Attributes: N-Z, 42

B

backColor menu attribute, 35
bgOffColor menu attribute, 35
bgOnColor menu attribute, 35
Border and Menu Item Areas, 11
Borders, 10
boxBorder menu attribute, 36
boxBorderColor menu attribute, 36

C

coding standards
 menus, 13
Colors, Fonts, Borders, Roll-overs, and
 Other Effects, 6
configuration parameters
 for menus, 31
 menuConfig.js, 13
Configuring Basic Menus, 17
Configuring the Menus for External
 Triggers, 21
Consistency, 6
Conventions, 3
conventions, typographic, 3

D

Defining the External Menu Triggers in
 the Page, 22

drop menu attribute, 36

E

elements
 a, 22
 div, 22
 head, 49
 img, 20
 menuItem, 18, 21, 28, 31, 39, 46
 menuSystem, 17, 31, 36, 40, 42, 44
 script, 49
Elements, 31
eTOffsetX menu attribute, 37
eTOffsetY menu attribute, 37
External Menu Trigger Code in the Page,
 22

F

fontColor menu attribute, 37
fontFamily menu attribute, 38
fontOnColor menu attribute, 38
fontSize menu attribute, 39
fontWeight menu attribute, 39
Full Width/Height Background Bar, 11

G

globalShiftX menu attribute, 39
globalShiftY menu attribute, 39

H

height menu attribute, 40
hideSelect menu attribute, 40
Hierarchy and Scope, 5
href menu attribute, 41

I

intBorder menu attribute, 41

J

JavaScript
 including for menus, 49
 JSP files
 applicationSettings.jspi, 13
 JSP Logic for Dynamic Menus, 27

L

labels
 menus, 13
 layertriggerPrefix attribute, 17
 layerTriggerPrefix menu attribute, 41
 localWidth menu attribute, 42

M

menu attributes
 arrowHref, 32
 arrowPadding, 33
 backColor, 35
 bgOffColor, 35
 bgOnColor, 35
 boxBorder, 36
 boxBorderColor, 36
 drop, 36
 eTOffsetX, 37
 eTOffsetY, 37
 fontColor, 37
 fontFamily, 38
 fontOnColor, 38
 fontSize, 39
 fontWeight, 39
 globalShiftX, 39
 globalShiftY, 39
 height, 40
 hideSelect, 40
 href, 41
 intBorder, 41
 layerTriggerPrefix, 17, 41
 localWidth, 42
 orientation, 42
 startX, 42
 startY, 43
 useInternalTriggers, 43
 width, 44
 Menu Coding, 13
 Menu Configuration for External Triggers, 21
 Menu Names, 13

Menu Standards, 5
 Menu System JavaScript Files, 50
 menuConfig.js file, 13
 menuFileURL variable, 14, 27
 menuItem
 attributes, 31
 element, 18, 31
 menus
 backgrounds, 18
 borders, 18
 coding standards, 13
 configuring, 50
 configuring individual items, 19
 configuring web site/application, 14, 17
 consistency, 6
 controlling primary item width, 25
 dynamically assigned, 27
 example outside the Global UI, 49
 externally triggered, 21-22
 fonts, 17
 hierarchy, 5
 implementing outside the architecture, 49
 including JavaScript library, 49
 multiple menuSystem elements, 25
 naming standards, 13
 placement on the page, 17
 scope, 6
 size, 17
 submenu hinting, 18
 turning on primary navigation, 13
 when to use, 5
 XML configuration markup, 31
 menuSystem
 attributes, 31
 element, 17, 31
 Multi-Line JavaScript Variable for Menus, 20
 Multiple Menu Systems, 25

O

orientation attribute, 42

P

Placing Menus Directly Under the Global Masthead, 13
 primary menu item width, 25

Purpose, Scope, and Audience, 3

S

Sample Blue Menu Configuration, 28
Sample Gold Menu Configuration, 29
Sample Menu Function Access by User Level, 28
Sample Silver Menu Configuration, 28
Simple Vertical Background Bar, 12
siteMenus variable, 13
Standard Colors, Fonts and Sizes, 7
startX menu attribute, 42
startY menu attribute, 43

T

Technical Syntax Issues, 19

U

useInternalTriggers menu attribute, 43
Using Externally Triggered Menus, 21
Using Multiple MenuSystem Elements, 25

V

Valid Attributes for menuSystem and menuItem, 31
variables
 menuFileURL, 14
 siteMenus, 13
Version 1.3, 47
Version 1.4, 46
Version 1.5, 1.6, and 1.7, 45

W

When to Use Menus, 5
width menu attribute, 44

X

XML Menu Configuration Markup, 31