GEPS User Interface (UI) Intranet Standards

Version: 1.0

Published: July 19, 2002

Author: UI Standards Team

GE Power Systems
e-Technology

Contents

Examples

Tables

Figures

Preface

Purpose, Scope, and Audience

This document is designed to supply both user-interface standards for Intranet web development and implementation instructions for the GEPS Global UI (user interface). Wherever possible, standards that affect some aspect of the global user interface are discussed along with specific implementation details.

This implementation does **not** cover the specific details of the Global UI within Epicentric, the portal server for GEPS. See the GEPS Global UI for Epicentric document on the e-Technology Intranet site for more information. The standards in this document still apply.

> **Note:** Intranet applications may also be deployed on the Extranet. The technology standards and tools presented in this document are also used for Extranet/Internet development. However, look and feel standards may be different.
>
> Projects for applications that may eventually move to the Extranet should also consider color and look and feel standards for the Extranet. Specific areas within this document may also highlight issues to consider for dual deployment. See the GEPS Extranet/Internet User Interface Standards for more detailed information.

Please send any comments, questions, or suggestions to GEPSeBizStds@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 or reference documents that may be of interest include:

- GEPS Intranet Menu Standards: for standards and implementation instructions on creating menus for web sites or applications using the XML Menuing System. This document also contains detailed reference information.
- 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.

- GEPS Global UI for Epicentric: for implementation instructions on creating web sites and menus with the GEPS Global UI in the Epicentric Portal Server.
- 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}

Chapter 1: The GEPS Global UI

The GEPS Global User-Interface (UI) includes a standardized architecture, standard templates (for primary pages and for pop-up window pages), standard cascading stylesheets (CSS), a standard menuing system, and other shared resources for use in Inside GEPS and its subsidiary web sites and applications. All new web sites and applications **must** use the Global UI; existing sites and applications will gradually be moved into the new user interface.

> **Note:** Web sites that should be deployed in the Epicentric Portal Server share the GEPS User Interface standards, but are implemented with a Global UI that is specific to Epicentric. See the GEPS Global UI for Epicentric document for implementation instructions.

The GEPS Global UI has been designed to accomplish the following goals:

- Provide a common 'look and feel' to allow visitors to Inside GEPS and all its subsidiary web sites and applications to have a sense of common community and to support ease of use through standard features and shared functionality.
- Provide a single source of code for the shared components of the Inside GEPS 'look and feel' elements. This includes the standard masthead, footer, CSS styles, and the look and function of menus.

> **Note:** This also insures a similar implementation with Extranet/Internet applications to allow Intranet applications to deploy on the Extranet with minimum effort.

- Provide simple and appropriate customizations of global components to support the various requirements of web sites and applications.

> **Note:** Standards within this document may be different for web sites and applications to accommodate their functional differences. See the definitions in the Glossary on page 75 that are marked *UIStd* to determine what standards apply to your development project.

- Allow easy updates to the global components.
- Provide a controlled mechanism to allow web sites and applications to upgrade to new versions of the global UI.

- Provide standard access to new, common functionality to simplify development. Examples include the standard menuing system and additional JavaScript libraries.
- Provide support for development environments that require local access to global elements and also support testing and production environments that must use a single, global source. (See the Moving Your Site or Application to Production section on page 11 for information on how this is handled.)
- Move towards new technology platforms (such as XHTML and JSP) for better maintenance and support of further goals (such as easier personalization and internationalization).

Installation, Implementation, and Deployment

Getting and Installing the Global UI

The GEPS Global UI is distributed to development groups as a ZIP file. The distribution for the GEPS Global UI is available from psintranet@ps.ge.com. If your web site should be deployed in the Epicentric Portal Server, see the GEPS Global UI for Epicentric document for installation and implementation information.

1. Decompress the contents of the ZIP file and place them in the appropriate location as shown below:
   - For web sites or applications using a J2EE application server being deployed as a WAR file (the standard), place the distribution under the war directory in the application.
   - For web sites or applications using a J2EE application server that are being deployed with an open directory (deprecated), place the distribution under the web application's root directory.
   - For non-JSP/J2EE web sites, place the distribution under the doc-root directory for your development web server.
2. Copy the applicationSettings.jspi, popupApplicationSettings.jspi, popupWindow.jsp, and menuConfig.js files from the /examples/intranet directory to the directory for web site or application pages.
3. Point the Global UI to your development server by changing the {serverPath} variable in applicationSettings.jspi and popupApplicationSettings.jspi files. See the Pointing to the Development Environment section on page 19 for more information.

For more information on the specific contents of this distribution, see the Table 1 on page 13.

Implementing the Global UI in Development

Once the Global UI is installed, you may configure the standard components of each web site or application page, within specific limits, including the masthead, the footer, and the menus. This configuration has multiple levels, as shown in the following illustration:

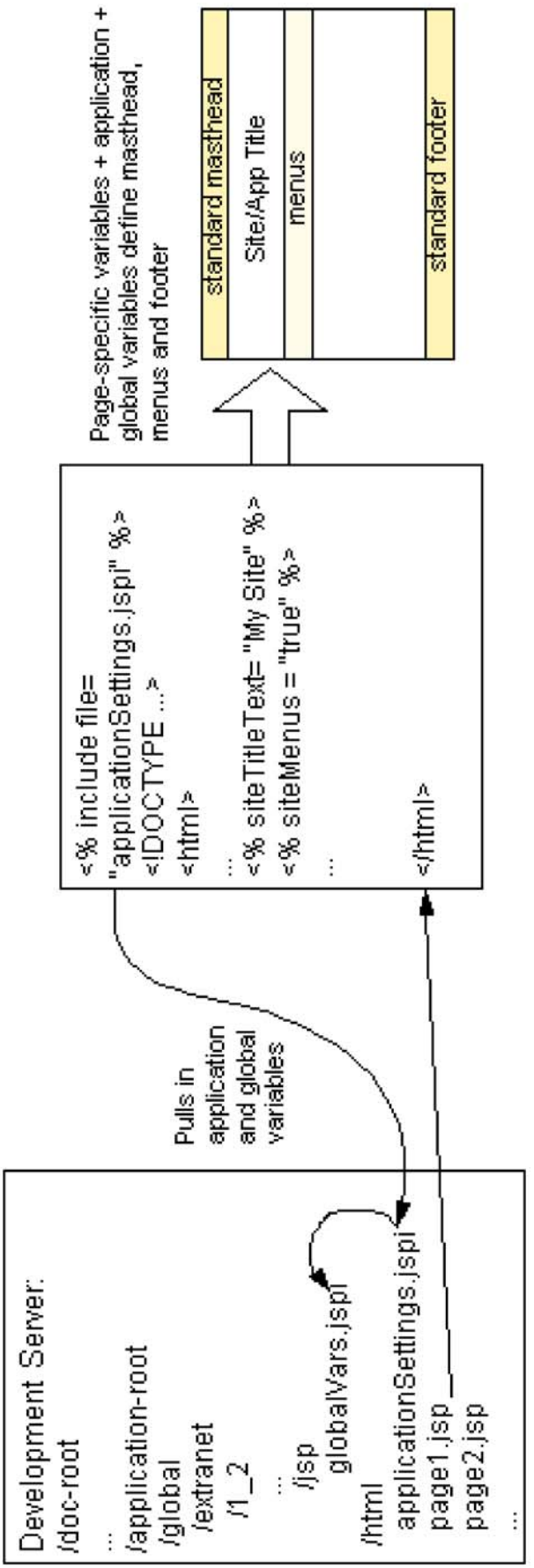

Figure 1. Configuring the Global UI In Development

Two sets of default configuration parameters are supplied in the GEPS Global UI Distribution: globalVars.jspi for primary pages and popupGlobalVars.jspi for pop-up pages. Do not change the defaults. Instead, use two site-specific levels of configuration to override the default configuration:

- **applicationSettings.jspi**: overrides the default configuration for all primary pages within the web site or application. This file also handles versioning to a specific 'look and feel' of Inside GEPS and also allows environment configuration such as pointing to the development server.
- **popupApplicationSettings.jspi**: overrides the default configuration for all pop-up pages within an application. Pop-up pages include secondary content such as help topics or other links that should open in the standard pop-up window.
- **each page**: each page may also set specific configuration parameters to unique values. For example, the help topic that should open from the Help button can be page-specific.

You include global components in each page using JSP. Standard CSS is used to supply required styles for text. In addition, web sites or applications can define menus as the primary navigation (up to three levels) for the pages within their scope. These menus are in addition to the global navigation for Inside GEPS.

Moving Your Site or Application to Production

Once the web site or application is ready to go to QA (and then move to production), there are a couple of steps that must be completed to insure that the local, development version of the Global UI is replaced with the shared, common version.

1. The development copy of the Global UI must be removed from an web-application directory tree.
2. Operations staff replace this local copy with a soft-link that points to the shared, global components.

This use of soft links allows many web applications and web sites to use one, shared Global UI. Information in the applicationSettings.jspi file for the web site or application identifies which version of the global components are used. This allows web sites and applications to migrate to new versions of the global UI in a controlled way, as shown here:

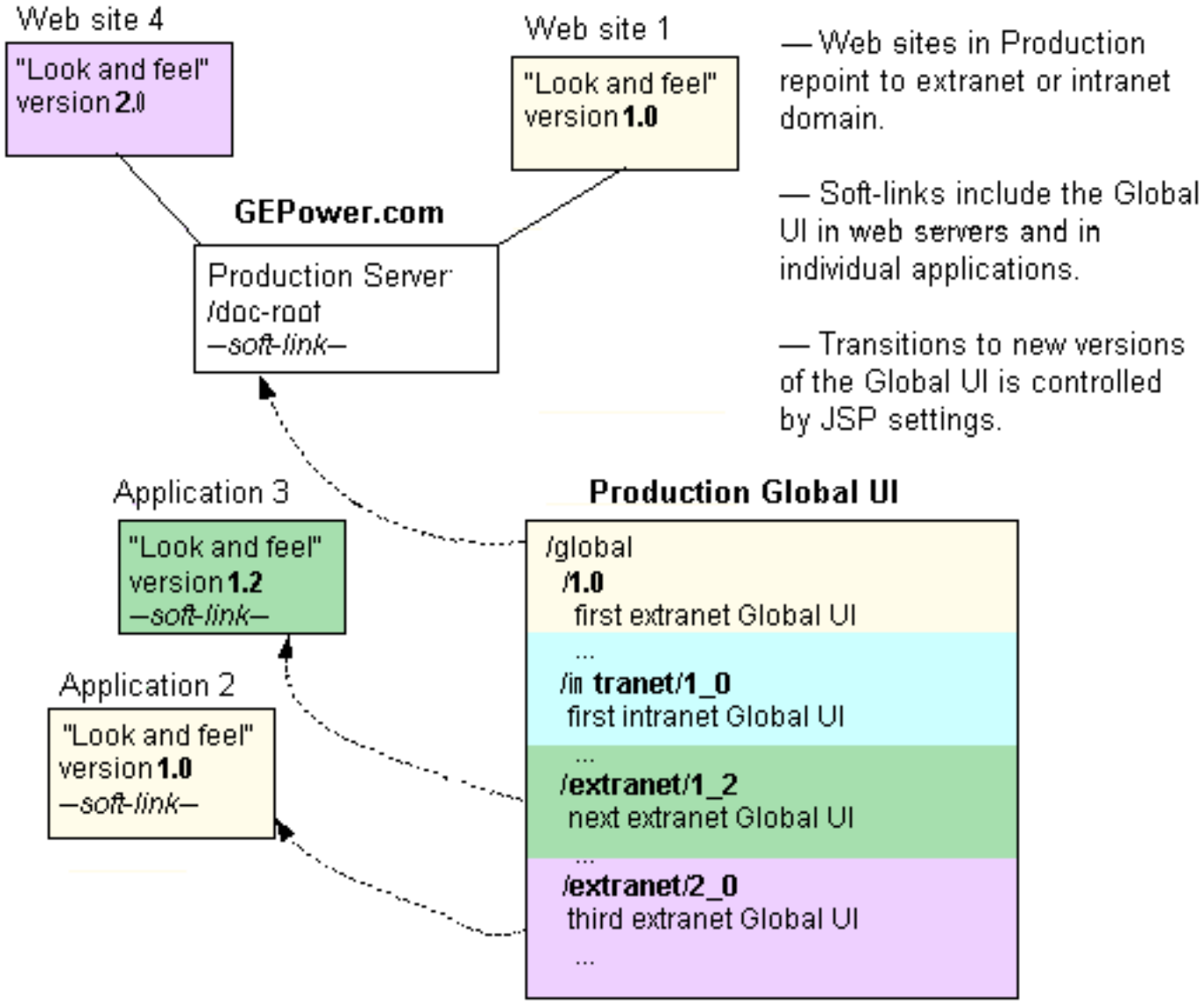

Figure 2. Pointing to the Global UI in Production

Static web sites that are not using JSP or J2EE point to the Inside GEPS domain to get the shared, global components. Dynamic web sites and applications **each** have individual soft links to include the global components.

GEPS Global UI Distribution

The distribution package for the Global UI includes the directories and files shown in Table 1 on page 13.

> **Important:** This version of the Global UI introduces separate templates for the Extranet and the Intranet. This change shows up as a separate directory under /global. Web sites or applications that are upgrading from previous versions of the Extranet Global UI must change path information for each *.jspi include file.

Your application should reside in a separate directory tree. For information on directory naming requirements, see the GEPS Java Development Standards. (Vendors may request a copy of this document from their GEPS IT Project Manager.)

Table 1. Global UI Distribution

Directory or File	Description
/global/extranet or /global/intranet	/global is the root directory for all shared content. /extranet contains the current version and all future versions of the Global UI for the Extranet. Similarly, /intranet contains all versions of the Global UI for the Intranet. For development, put this directory off the doc-root directory of your web server or, for J2EE web-applications, off the web-application-root directory. In production, this uses the shared content from the Inside GEPS domain. **Warning:** Do **not** modify any content in this directory or its subdirectories.
/*n_n*	The directory tree for a specific version of either the Extranet or the Intranet Global UI. For example, **/global/extranet/1_2** is version 1.2 of the Extranet Global UI.
/css	Contains the standard, shared CSS stylesheets for use in all Inside GEPS web sites and applications for that version of the Global UI.
/img	Contains the shared graphics, such as the Inside GEPS logo that version of the Global UI.

Table 1. Global UI Distribution (continued)

Directory or File	Description
/js	Contains the shared JavaScript libraries that enable the XML menus, as well as providing common functionality for all web sites and applications for that version of the Global UI.
/jsp	Contains the shared JSP code for use in applications and web sites for that version of the Global UI.
/examples/extranet or /examples/intranet	Contains example files that you can use as templates or to help illustrate some of the specific configuration that you may need to do.
/doc/extranet or /doc/intranet	Contains the documentation for this distribution for the Extranet/Internet templates or the Intranet templates respectively.

Chapter 2: Global Templates and JSP

Using the Global Templates

Five steps must be applied to each web site or application page to implement the standard header and footer, the standard stylesheets, and the XML menus:

1. **Use the JSP Template on Each Primary Page:** this includes all of the JSP elements that pull in the standard styles, standard JavaScript libraries, and the masthead and footer.

> **Note:** If your application must use framesets, see the Using Frames in Primary Pages section on page 26. Framesets are deprecated and generally should **only** be used if third-party vendor products that the application needs require them.

An example of the code for this template is shown below:

Example 1. Primary JSP Page Template

```
<%@ include file="applicationSettings.jspi" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>title for this page</title>
<%@ include file="/global/intranet/1_0/jsp/headConfig.jspi"%>
</head>
<body <%=bodyAttributes%>>

<%@ include file="/global/intranet/1_0/jsp/header.jspi"%>

put your content here

<%@ include file="/global/intranet/1_0/jsp/footer.jspi"%>

</body>
</html>
```

This example assumes that the applicationSettings.jspi file is located in the same directory as all of the primary pages of your application. If you organize primary pages into subdirectories, you must change the path information in the <%@ include file="*path-to-settings-file*"%> tag within each page.

2. **Use the Pop-up JSP Template for Pages using the Standard Pop-up Window:** this window **must** be used for online help for applications or web sites. Web sites may use other pop-up windows but these should be limited (use them sparingly). Applications may use pop-up windows (standard or otherwise) for the following types of content:

   - Online help (from the masthead link, must use standard pop-up)

   > **Note:** If help is being written by a technical writer using the GEPS XML Help application (in XMetaL), these pages are generated automatically. See the Connecting Online Help section on page 33 for the additional setup needed to connect the help to the application.

   - Site maps (from the masthead icon)
   - Tutorials (application-specific)

   Secured content should generally not appear in pop-up windows as this forces the user to log in again.

   An example of the code for the standard pop-up template is shown below:

Example 2. Standard Pop-up JSP Page Template

```
<%
  String popupType = "privacy";
  String thisName = "my-privacy.jsp";
  String mainContent = "";
  if (request.getParameter("popupType") != null)
  {
     popupType = request.getParameter("popupType");
  }
%>
<%@ include file="popupApplicationSettings.jspi" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>=
<title>Popup Window Content Page</title>
<%@ include file="/global/intranet/1_0/jsp/popupHeadConfig.jspi" %>
```

Example 2. Standard Pop-up JSP Page Template (continued)

```
</head>

<body <%=bodyAttributes%>>
<%@ include file="/global/intranet/1_0/jsp/popupHeader.jspi" %>

<form name="printForm" action="<%= thisName %>" method="post">
<input type="hidden" name="placeHolder" value="placeHolder">
</form>

<%
  if (request.getMethod().equalsIgnoreCase("post"))
  {
%>
<script>
  printMe();
</script>
<%
  }
%>

<!-- table to ensure font is set correctly on all browsers -->
<table border="0" cellpadding="5" cellspacing="0" width="100%">
  <tr>
   <td align="left" valign="top" width="100%">

<!-- Popup content goes here -->

   </td>
  </tr>
 </table>
<!-- End of table to ensure font is set correctly on all browsers -->

<%@ include file="/global/intranet/1_0/jsp/popupFooter.jspi"%>

</body>
</html>
```

The {popupType} and {thisName} variables change, depending on what the link to the pop-up content is associated with and the name of the pop-up page (respectively). In addition, the path in the <%@ include file="*path-to-pop-up-settings-file*" %> tag may also need to change if your pop-up pages are in different directories from the popupApplicationSettings.jspi file.

3. **Add Content to the Page:** see the following sections for implementation information, coding standards, styles, and examples for page content:

   - Text Elements section on page 59
   - Tabs section on page 62
   - Buttons and Links section on page 64
   - Fields section on page 67
   - Selection and Sorting Elements section on page 68
   - Tables section on page 69
   - Pop-up Windows section on page 72

   You may also use any of the standard JavaScript Library functions within your pages. See the GEPS Standard JavaScript Library Reference for information.

4. **Configure JSP Variables:** the applicationSettings.jspi and popupApplicationSettings.jspi files in the distribution contain variables that may need to be changed to properly implement various aspects of primary or pop-up pages, the standard masthead, or the footer. You may also set individual variables within any page for page-specific needs. See the Configuring JSP Variables for Primary Pages section on page 18, Configuring JSP Variables for Pop-up Pages section on page 33, and the Setting JSP Variables in Individual Pages section on page 35 for details.

5. **Configure the Menus for Your Site:** global navigation for Inside GEPS is predefined and not configurable. You may define menus for your individual web site or application by copying the menuConfig.js file from the /examples/intranet directory and placing it in the directory tree for your web site.

   Then change the XML configuration information in your local menu configuration file. See the GEPS Intranet Standard Menus document for menu design principles and instructions on how to implement your menus.

Configuring JSP Variables for Primary Pages

The applicationSettings.jspi file in /examples/intranet of the GEPS Global UI distribution controls the variables to define the standard masthead, web site/application title, menus, and footer for primary pages (basically any page that does not belong in a pop-up window). Generally, you should use this file to define aspects of the standard page elements that are commonly true for most of your web site or application but are different from the global values.

> **Important:** You should remove any variable definitions in this local file that do **not** differ from the global values.

The variables control the content of each area of the masthead and footer, as shown in Figure 3 on page 19. Most elements of the masthead and footer are required, but others are optional.

Figure 3. Masthead and Footer Areas

The masthead area has three separate 'subareas' as shown in Figure 4 on page 19.

- The menu areas supply both global navigation for Inside GEPS and site- or application-specific navigation. This sample shows horizontal site menus, but they may also be configured to display as a left vertical bar.
- The user assistance and user access area contains buttons to assist users and also displays security and user access buttons for Single Sign-on.
- The Branding provides basic identification and branding and the Inside GEPS graphic can also be set as a 'home' button to return users to the Inside GEPS home page.

Figure 4. Masthead Subareas

Pointing to the Development Environment

In addition to controlling aspects of the masthead and footer, applicationSettings.jspi contains one variable, {serverPath}, that you use during development to point to the local copy of the Global UI. Set this during development and then remove the variable definition prior to delivering the web site or application to Quality Assurance.

{serverPath}

This variable determines the path to the server. In development environments, set this variable to the IP address or domain name of your development web server.

In production, you must **remove this variable** so that the global value is used.

Setting Global Navigation and Site Title Variables

These variables control the presence of the global menus and the linking for the Inside GEPS logo. They also control presence of the standard masthead graphic bar and the title for the web site or application that appears in the standard masthead. These features are required for web sites and applications.

> **Note:** Applications that have technical, functional, or usability issues with including these features have specific options for these requirements covered in the Application Options to Control Global Navigation/Shutdown section on page 29.

{globalMenus}

This variable controls whether the global menu (top menu bar) is shown in the standard masthead.

May be true of false. The global default is true.

Web sites cannot change this variable. Application options to change this variable or control its effect are covered in the Application Options to Control Global Navigation/Shutdown section on page 29. If the application uses framesets, see also the Using Frames in Primary Pages section on page 26.

{globalHeader}

This variable controls whether the standard graphics bar, including the application or web site title, is shown in the standard masthead. If this is set to false, the standard graphics bar is surpressed and the {clickableLogo}, {clickableLogoURL}, {siteTitleText}, {gePower} and {search} variables are ignored.

May be true or false. The global default is true.

> **Important:** Web sites or applications may **only** turn off the graphic bar in the masthead (i.e., set the variable to false) if they obtain an exception in ePMM Toll Gate 2. Exceptions are rarely granted, usually in situations where a web site or application is connected to and part of a corporate-wide project and must fit the corporate standard.

{clickableLogo}

This variable controls whether or not the Inside GEPS logo in the standard masthead acts as a link to the Inside GEPS home page. This variable is ignored if the {globalHeader} variable is set to false.

May be true or false. The global default is true.

Web sites cannot change this variable. Application options to change this variable or control its effect are covered in the Application Options to Control Global Navigation/Shutdown section on page 29.

{clickableLogoURL}

This variable only takes effect if the {clickableLogo} variable is set to true. The global default is to link to the Inside GEPS home page. This variable is ignored if the {globalHeader} variable is set to false.

The default value for this variable should not be changed.

{siteTitleText}

This variable contains the text that appears as the web site or application title in the standard masthead. It is required for all web sites and applications. This variable is ignored if the {globalHeader} variable is set to false.

The global default is "GEPS Intranet". Change this to the title for your web site or application.

Setting User Access Variables

User access variables control whether buttons for Single Sign-on are activated. Single Sign-on is the GE-wide application that handles user authentication and authorization.

{userAccess}

This variable determines whether user access buttons appear in the standard masthead.

May be true or false. The global default is false which prevents any of the user access buttons from appearing.

If you set this to true, the {userAccessApproved} and {userAccessLogin} variables take effect. If this is set to false, no user access buttons appear in the standard masthead.

{userAccessApproved}

This variable controls which user access buttons appear within the standard masthead. Set this to true on pages where the user must have access approval to reach the page. This displays the Logout button.

May be true or false. This variable only takes effect if the {userAccess} variable is set to true. The global default is false.

If this variable is set to false, the Login button may still appear depending on the {userAccessLogin} variable.

{userAccessLogin}

This variable controls whether or not the Login button appears in the standard masthead. This should only be set to true for splash pages to applications that are secured or for web site pages that lead to secure sections of the site. If you set this variable to true, you **must** also set the {loginLink} variable.

May be true or false. This variable only takes effect if the {userAccess} variable is set to true and the {userAccessApproved} variable is set to false. The global default is false.

{loginLink}

This variable is only active if the {userAccessLogin} variable is true and should only be set on pages that lead to secured areas of an application or web site.

> **Important:** The default value for the variable simply opens a JavaScript alert window stating that Login is not currently available. If you enable the Login button, you **must** set this variable in every page where Login appears.

To activate the Login button, you must enter the URL for a secured page within the application in this variable. This is the page that users will be redirected to upon a successful login. In addition, the secured page must be set up in SiteMinder and the application must be configured to integrate with Single Sign-on. For more information on the required set up, see the Single Sign-on Implementation Guide.

Setting User Assistance Variables

User assistance variables control buttons or other utilities that allow a user to find additional information, make useful contacts, or quickly navigate to common areas.

Global User Assistance

{gePower}

This variable controls whether or not the **gepower.com** button appears in the graphics bar of the standard masthead. The global default is true.

> **Note:** This variable is ignored if the {globalHeader} variable is set to false (the entire graphics bar in the standard masthead is surpressed).

Web sites cannot change this variable. Application options to change this variable or control its effect are covered in the Application Options to Control Global Navigation/Shutdown section on page 29.

{search}

This variable controls whether or not the **Search** button appears in the graphics bar of the standard masthead. May be true or false. The global default is true.

> **Note:** This variable is ignored if the {globalHeader} variable is set to false (the entire graphics bar in the standard masthead is surpressed).

Web sites cannot change this variable. Application options to change this variable or control its effect are covered in the Application Options to Control Global Navigation/Shutdown section on page 29.

{gepsHome}

This variable controls whether or not the **GEPS Home** button appears in the graphics bar of the standard masthead. May be true or false. The global default is true.

> **Note:** This variable is ignored if the {globalHeader} variable is set to false (the entire graphics bar in the standard masthead is surpressed).

Web sites cannot change this variable. Application options to change this variable or control its effect are covered in the Application Options to Control Global Navigation/Shutdown section on page 29.

Site-Specific User Assistance

These variables supply common user assistance buttons that are **strongly** recommended, but not required in all situations.

{siteHomeButton}

This variable controls whether or not the **Home** button appears in the standard masthead. This button should return users to the web site or application home page.

May be true or false. The global default is true.

{siteHomeButtonAlt}

This variable only takes effect if the {siteHomeButton} variable is set to true. Enter the text that should display for this button if the user has turned off graphics.

{siteHomeButton-Link}

This variable only takes effect if the {siteHomeButton} variable is set to true. Enter the URL for the home page of the web site or application. The default value is simply a JavaScript alert indicating that you must set this variable locally.

{siteHomeButton-Mouse}

This variable only takes effect if the {siteHomeButton} variable is set to true. Enter the text that should appear in 'tool tips' when the mouse hovers over the 'home' button.

{siteMapButton}

This variable controls whether or not the Site Map button appears in the standard masthead. May be true or false. The global default is false.

{siteMapURL}

This variable determines the URL or JavaScript function to execute from the Site Map button in the standard masthead. The global default simply opens a JavaScript alert box indicating that this variable should be reset.

Site maps are **recommended** but optional for web sites or applications. Applications may open site maps in a pop-up window to avoid technical or functional problems.

{helpButton}

This variable determines whether or not the Help button appears in the standard masthead. Help is optional for web sites but **strongly recommended** for applications.

May be true or false. The global default is false.

{helpButtonLink}

This variable determines the URL or JavaScript function to execute from the help button in the standard masthead. For applications, help **must** open in the standard pop-up window using the following function:

```
"javascript:popupWindow('popupWindow.jsp',
'help','help/toc.jsp');"
```

The topic to open from this button **should** be context-sensitive for applications (the help page that opens depends on the current page). Because of this, this variable is almost always set on individual pages of the application. The variable in this file defines a default help topic, usually the table of contents. Change the last parameter in the function to the URL to identify the table of contents page for help.

Setting Site/Application Menu Variables

These variables determine whether site-specific menus should be used. Web sites are **strongly encouraged** to use menus for their primary navigation, although other navigation paradigms are allowed. Applications should use menus, if that paradigm is appropriate for their functional flow, but other paradigms are allowed. See the GEPS Intranet Menu Standards for more information about menu standard, colors, etc.

{siteMenus}

This variable controls whether site or application menus using the XML Menuing system display on the page. Applications or web sites that are not using menus for their functional flow may set this to false. Web sites and any applications that use menus should set this to true.

May be true or false. The global default is true.

{menuFileURL}

This variable determines the path and file name for the configuration file with the menus for your web site or application. The global default is menuConfig.js.

Use this variable if your web site or application has multiple menu configuration files or needs to define menus dynamically. For more information on configuring menus or setting them dynamically, see the GEPS Intranet Menu Standards document.

{mastheadSiteMenus}

Determines whether the default horizontal menu bar displays in the masthead. May be true or false. The default value is true. Change this variable to false if your site or application menus display vertically or use a custom background bar.

If this variable is true, the global masthead includes an additional horizontal bar using a custom drop shadow background where horizontal site menus should display. See the GEPS Intranet Menu Standards for specific configuration information to allow horizontal menus to display within this masthead area.

Setting Footer Variables

These variables control the various features on the standard footer.

{contactInfoButton}

This variable determines whether or not the Gatekeeper link appears in the standard footer. May be true or false. The global default is true.

{contactInfoButton-Link}

This variable only takes effect if the {contactInfoButton} variable is set to true. Enter the email address for the Gatekeeper link in the standard footer. The default contact is the GEPS Intranet Business Owner.

{contactInfoName}

Enter the name that should appear in the standard footer as the Gatekeeper for this web site or application. The default value is the GEPS Intranet Business Owner. .

{webmasterButton}

This variable determines whether or not the Gatekeeper link appears in the standard footer. May be true or false. The global default is true.

{webmasterButton-Link}

This variable only takes effect if the {webmasterButton} variable is set to true. Enter the email address for the Webmaster link in the standard footer. The default contact is the GEPS Intranet IT Project Manager.

{webmasterName}

Enter the name that should appear in the standard footer as the Webmaster for this web site or application. The default contact is the GEPS Intranet IT Project Manager.

Using Frames in Primary Pages

Although frames are a deprecated technology for GEPS web applications, they are required in some cases. For example, frame are allowed if a DHTML solution is not viable or in situations where third-party vendor software is used that requires frames.

To implement the Global UI in applications that use frames, you must:

1. **Implement three standard templates** for the frame set, the masthead frame, and the footer frame. These templates only need to

be implemented once, **unless** some pages use different components within the masthead causing the height of the masthead frame to vary.

Example 3. Frame Set Template

```
<%@ include file="applicationSettings.jspi" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
        "http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd">
<html>
 <head>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1"/>
  <title><%=siteTitleText%></title>
  <%@ include file="/global/intranet/1_0/jsp/frameHeightConfig.jspi" %>
 </head>
 <frameset rows="<%=headerFrameHeight%>,*,50" frameborder="0">
  <frame src="framesetHeader.jsp" name="globalHeader" marginheight="0"
    marginwidth="0" scrolling="no" noresize="noresize" />
  <frame src="application content" name="appFrame"
    noresize="noresize" marginheight="0" marginwidth="0" />
  <frame src="framesetFooter.jsp" name="globalFooter" marginheight="0"
    marginwidth="0" scrolling="no" noresize="noresize" />
 </frameset>
</html>
```

If the application is using one configuration for all masthead JSP variables, the height of the frame for the standard masthead is determined automatically and set in the {headerFrameHeight} variable (see next step). If the masthead varies within the application, then each variation must have its own frameset document and you may need to reset the {headerFrameHeight} variable for each variation.

The first and last frame elements are required. The middle frame element can be replaced by any number of frame elements that define the configuration for application content.

Example 4. Global Masthead Frame Template

```
<%@ include file="applicationSettings.jspi" %>
<%
  isFrameset = true;
  menuFileURL = "framesetMenuConfig.js";
  siteTitleText = "Frameset Example Page";
%>
```

Example 4. Global Masthead Frame Template (continued)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
 <head>
  <title><%=siteTitleText%></title>
  <%@ include file="/global/intranet/1_0/jsp/headConfig.jspi"%>
 </head>
<body <%=bodyAttributes%>>
 <%@ include file="/global/intranet/1_0/jsp/header.jspi"%>
 <%if((userAgentIsNS4)&&((globalMenus)||(siteMenus))){%>
  <script language="javascript1.2" type="text/javascript">
   writeAllMenuDivs();
  </script>
 <%}%>
</body>
</html>
```

Example 5. Global Footer Frame Template

```
<%@ include file="applicationSettings.jspi" %>
<% globalMenus = false; %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
 <head>
  <%@ include file="/global/intranet/1_0/jsp/headConfig.jspi"%>
  <title><%=siteTitleText%></title>
 </head>
<body <%=bodyAttributes%>>
 <%@ include file="/global/intranet/1_0/jsp/footer.jspi"%>
</body>
</html>
```

---

2. Set two JSP variables, in addition to all the standard JSP variables that control the standard masthead and footer content:

   isFrameset

   This variable defines whether frames are in use. If set to true, it repoints the menu configuration file used for the global menus

to one that works within the masthead frame. The default value is false. This variable must be set in the frame holding the standard masthead (see Example 4 on page 27).

> **Note:** When frames are used, drop down menus do not perform reliably. Because of this, both the global menus and site menus, if any, **cannot** have more than one, top level (acting as buttons on a toolbar rather than as a menu).

headerFrameHeight

This variable determines the height to set for the frame that contains the standard masthead if the application is using framesets. The value for this variable is automatically calculated based on which components the application is using from the standard masthead.

If some pages within the application use different components of the standard masthead, the frame height for the masthead may need to be different. To accomodate this, you must:

- Define different frame set documents for each variation of the masthead that the application is using.
- Define individual values for the frame height in each frame set document using this variable. This **must** be redefined after the include directive for frameHeightConfig.jspi.

3. **Write application pages as XHTML or JSP to fill the application areas of the frame set**. These pages generally do not need to contain any JSP code for the Global UI as the masthead and footer are handled in other frame set files. See the /examples/intranet/framesetAppSpace.html file for an example. Links and buttons are then targeted for the application frame (or frames).
4. Configure site menus with a single level. Drop down menus do not perform reliably in conjunction with frames so only one level is supported.

   See the example menu configuration file, /example/intranet/frameset-MenuConfig.js, for an example of how to configure menus within frames. See the GEPS Intranet Menu Standards for more information on menu configuration.

Application Options to Control Global Navigation/Shutdown

Global navigation elements in the standard masthead are required. However, applications can sometimes run into technical, functional, or usability issues with including many of these elements . In these cases,

applications have the following options for handling these issues (in order of preference):

- Have the application open in its own pop-up window and turn off any problematic global navigation elements. This leaves the primary browser window open so users can still access the Intranet, but leaves the application in complete control of shutdown issues.

  > **Note:** One downside to this choice is that any users who have already logged in must log in again if the application is secured (through Single Sign-on).
  >
  > Applications that will also be deployed on the Extranet should also be aware that they are **not** allowed to open in separate windows on the Extranet.

  Applications that open in separate windows must still follow standards including:

  – Fonts, colors, links, and general behavior.
  – Coding standards and use of the standard menus.
  – Retaining all aspects of the masthead and footer that are not a problem.
  – Including a standard mechanism (such as an "Exit" or "Close" button) to allow users to easily exit the application and close the window.

- Applications may turn off any of the following global navigation elements in the primary browser window, with one additional requirement:

  – Global Menus ({globalMenus})
  – Inside GEPS Logo ({clickableLogo})
  – Inside GEPS Home {{gepsHome}}
  – GEPower.com button ({gepower})
  – Search button ({search})

  If the application turns off global navigation elements in the primary browser window, it **must** implement an Exit button on every page of the application. This Exit button **must** return the user to either the Inside PS home page or to the page from which the application launched where global navigation elements are present.

  Applications that will also be deployed on the Extranet should also be aware that they are **only** allowed to turn off Extranet global navigation elements if they request an exception to these requirements and it is granted during e-PMM Toll Gate 2 reviews.

- Enable exit prompting for the global navigation elements and add shutdown logic to handle any session, state, or data integrity issues. See the Enabling Exit Prompting and Activating Shutdown Logic section on page 31 for details.

Enabling Exit Prompting and Activating Shutdown Logic

To control the effects of a user exiting an application through a global navigation element, you must:

1. **Enable exit prompting** for the appropriate global navigation elements in applicationSettings.jspi.

   Exit prompting uses the doExit JavaScript function to prompt the user to confirm an action that exits the application, indicating that this may have consequences the user may not have considered. Shutdown logic for the application can be hooked into this to ensure proper clean up if the user confirms that they do want to exit the application.

   To enable exit prompting, you assign the doExit function to JSP variables for the global navigation elements listed in Table 2 on page 31.

Table 2. JSP Variables to Set for Exit Prompting

Global Navigation Element	JSP Variable and Exit Prompt Value
Inside GEPS Logo	{clickableLogoURL} = "Javascript:doExit ('http://ps.home.ge.com/')";
Inside GEPS Home button	{gepsHome} = "Javascript:doExit('http://ps.home.ge.com/')";
Global Menu	{globalMenuFile}= "globalMenuConfigExitHook.js"; This variable changes the configuration file for the menu to one that handles exit prompting for each menu item.
GEPower.com button	{gePowerURL} = "Javascript:doExit('http://www.gepower.com")";
Inside GEPS Search button	{searchURL} = "Javascript:doExit('http://web1.geps.ge.com/search/')";

2. **Create a shutdown JSP page** with the required application logic.

   Shutdown pages must include all clean up logic within a JSP tag. An example of the final required code to return the user to the actual exit destination page is shown below:

Example 6. Shutdown Page for Exit Prompting

```
<%
   // Do some app shutdown stuff here

   String realpath = request.getParameter( "hook_url" );

%>
<script language="javascript1.2" type="text/javascript">
   window.location = "<%=realpath%>";
</script>
```

In this example, the value of the hook_url parameter is set from the URL specified in the doExit JavaScript function assigned to the appropriate JSP variable.

3. **Activate the shutdown page** using two JSP variables.

{hookShutdown}

This variable determines whether or not to execute some final cleanup logic before users exit an application through a global navigation element (menu item, link or button) that has exit prompting enabled through the doExit function.

May be true or false. The global default is false.

If this variable is set to true, any link or button that executes the doExit function prompts the user to confirm that they want to exit the application. If the user confirms this prompt, the JSP page identified in the {hookShutdownURL} variable is executed before returning the user to the URL specified for that link or button.

{hookShutdownURL}

This variable only takes affect if the {hookShutdown} variable is set to true and exit prompting has been enabled for a global global navigation element (menu item, link or button). Enter the URL of the intermediate JSP page (with clean up logic) that must execute from that menu item, link or button before returning the user to the final destination.

The final destination URL (where the user is returned after intermediate cleanup is completed) is determined by the URL parameter passed in the doExit function assigned to the global navigation element. See the Application Options to Control Global Navigation / Shutdown section on page 29 for an example

of the required code for an intermediate JSP page used for shutdown.

Configuring JSP Variables for Pop-up Pages

Web sites should use the standard pop-up window for online help content and simple pop-up windows for rare situations for large graphics (see the Oversized Graphics section on page 42). No other web site content should appear in pop-up windows. Applications use the standard pop-up window for content from the following elements within primary pages:

- Contact Us (from the primary masthead)
- Feedback (from the primary masthead)
- Help (from the primary masthead)
- Privacy Policy (from the standard footer)
- Terms of Use (from the standard footer)

Configuration During Development

The popupApplicationSettings.jspi file in /examples/intranet directory of the GEPS Global UI distribution controls the variables to define the standard masthead and page title for pages that use the standard pop-up window. This file also uses the {serverPath} variable during development to point to the local copy of the Global UI. The variable should be removed prior to dropping code in QA.

{serverPath}

This variable determines the path to the server. In development environments, set this variable to the IP address or domain name of your development web server.

In production, you must **remove this variable** so that the global value is used.

Connecting Online Help

Online help for web sites or applications that has been authored using the GEPS XMetaL Help application creates standard pop-up window pages for help automatically. Integrating these online help pages into a web site or application requires these minimal steps:

1. Notify the technical writer with the information listed here. This information is used in the GEPS XMetaL Help authoring system to generate the correct versions and styles of pop-up pages.
   - Is this an Intranet or Extranet application or site?
   - Which version of the Global UI is being used?
   - What is the relative path, starting from the web-application root directory to the popupApplicationSettings.jspi file?

2. Set the {serverPath} variable in popupApplicationSettings.jspi to the development server.
3. Configure the page title and buttons of the standard pop-up window. See the Configuring Page Titles and Buttons section on page 34 for information.
4. Use the topic map supplied by the technical writer to add {helpButtonLink} variables to the web site or application pages so that each context-sensitive help topic is connected to the correct page.

Configuring Page Titles and Buttons

The popupApplicationSettings.jspi file repeats the variables shown below for **each category of pop-up content** to allow you to define different page titles and buttons for different categories of pop-up window content. You can also define these variables on individual pages of the content to override the settings of this file.

The window supports up to three definable buttons for each category of content. Generally, these buttons are used to supply navigation within the content for that type. A typical example of this is have a Table of Contents and an Index button for online help content. If you use buttons within the standard pop-up window, you must configure them sequentially (i.e., you cannot define button 1 and button 3 but not button 2).

{buttonBarVisible}

May be true or false. If you set this to true, up to three buttons may appear on the content masthead in the pop-up window for this type of pop-up content. If this is true, you must also set {button1Bar} to true.

{button1Bar}, {button2Bar}, and {button3Bar}

May be true or false. These variables only takes effect if the {buttonBarVisible} variable is set to true.

These variables controls whether the first, second, or third button appears on the content masthead in the pop-up window for this type of pop-up content. If one of these variables is true, you must also set the corresponding {button*n*Text} and {button*n*Link} variables.

{button1Text}, {button2Text}, and {button3Text}

These variables only takes effect if the {buttonBarVisible} and respective {button*n*Bar} variables are both set to true. They define the labels for the first, second, or third button in the content masthead for this type of pop-up content.

> **Caution:** The horizontal space on the content masthead of the pop-up window is **very** limited (default width is 450 pixels including the title). Button names should be limited to a single short word, if at all possible.

{button1Link}, {button2Link}, and {button1Link}

These variables only takes effect if the {buttonBarVisible} and the respective {button*n*Bar} variables are both set to true. They defines the URLs for the content to display in the pop-up window for this type of pop-up content when Button 1, 2 or 3 is clicked.

{mainContent}

The URL for default content to display in the pop-up window for this type of pop-up content if the window is opened without a page specified.

{popupTitleText}

The text to appear as the page title in the content masthead of the pop-up window for this type of pop-up content.

> **Note:** Horizontal space in the content masthead is **very limited** (default width is 450 pixels including buttons). Page titles should be short, especially if buttons are also used. If the text of the page title is too long to fit within the available width, the page width extends causing a horizontal scroll bar to appear.

Setting JSP Variables in Individual Pages

You may also set specific variables in individual pages of the web site or application to override default values. In some cases, specific variables **should** be different for each page, such as the {helpButtonLink} variable in applications. Help links for applications should be context-sensitive, so the help URL would be different for each page of the application.

Variables should be set within the head element of the page and **before** any other JSP includes. The example shown below resets both the {siteTitleText} and {helpButtonLink} variables for a primary page:

Example 7. Setting JSP Variables in Individual Pages

```
<%@ include file="applicationSettings.jspi" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Update Profile Page</title>
<% siteTitleText="Update Your Profile" %>
<% helpButtonLink="javascript:popupWindow('popupWindow.jsp','help','help/prfupd.jsp');" %>
<%@ include file="/global/intranet/1_0/jsp/headConfig.jspi"%>
</head>
<body <%=bodyAttributes%>>
<%@ include file="/global/intranet/1_0/jsp/header.jspi"%>

  put your content here

<%@ include file="/global/intranet/1_0/jsp/footer.jspi"%>
</body>
</html>
```

Chapter 3: Common Standards

Platforms

This section outlines the fundamental assumptions about wired client devices (not wireless), technology capabilities, and the implementation that web sites and applications must support or are limited to.

Table 3. Client Platforms and Technology

Operating Systems	Any of the following operating systems that the supported browsers can run on: • Windows (32-bit only) • UNIX • Mac • Linux
Screen Resolution and Window Size	• 800 x 600 pixels fixed width (**note**: actual width limit is 759 pixels to allow for scroll bars, etc.) is the minimum resolution and window size that should be supported. Wherever possible, all required interactive elements (menus, buttons, fields, etc.) for a page must fit within this space. • Horizontal scrolling is **not** allowed on pages unless the user resizes the window below the minimum width. • Pages **must** be designed to fill the width of the window when users resize the browser window larger. Simply adding white space in the margins is generally not an acceptable design to meet this requirement.
Color Support	Support for millions of colors is assumed. 'Web safe' colors are preferred, but **not** required. Colors should be taken from the approved Color Palettes section on page 40. Pages with non-'web-safe' colors should be tested carefully for cross-browser rendering issues.
Graphics	GIF and JPG. See the Graphics section on page 42.

Common Design Standards

Design Criteria

The GEPS Global UI has been designed to meet several fundamental design criteria:

- Global elements of the user interface do not contain any horizontal constraints (left or right side). Intranet web sites and applications that will never be moved to the Extranet may have vertical navigation (such as left vertical menus). Applications that are likely to move to the Extranet may **only** use horizontal menus.
- Global elements and all page content should use the full width of the browser window (800 pixels759 pixels). Content should resize to larger widths gracefully and not use white space as a 'filler'.
- The approved font list is designed to work on all supported platforms.
- Readability and good balance in text requires a limited use of highlighting techniques, such as color or bold effect. Italics and underlines should be avoided to enhance readability and limit user confusion (underlines are frequently associated with links).

Color Palettes

The color palette for Inside GEPS is shown below. Applications and web sites should use the hexadecimal identifiers for these colors.

Applications and web sites may change the saturation or values of these colors to obtain sufficient variety for graphic elements. Colors must use these hues or hues that are complimentary (180 degrees opposite).

Text colors are determined by the global CSS stylesheet. See the CSS Coding section on page 51 for more information.

Graphics

Maximum Size

Graphics files must be small enough that, combined with other content in a page, the performance objectives (for static pages) are met. One simple way to calculate download time for the page is:

```
page size < = performance seconds x 5
```

Keep graphics small and include the height and width in coding. You can use Adobe Photoshop's Image Ready tool to compress graphics.

Colors

Graphic colors should balance well with the color palette used in the Global UI. Intranet applications that may be deployed on the Extranet must take **special** care in designing graphics and choosing colors that will work well in both environments.

Oversized Graphics

Large graphics, such as graphs or charts, that do not fit within the standard supported window size (800 x 600 pixels) are sometimes required within content. Web sites and applications may use a 'thumbnail' version of oversized graphics with a linking caption of 'click to enlarge.' This caption should be the **only** link to the oversized graphic.

The link should open a simple pop-up window with the following characteristics:

- The window size must open **at less** than 640 x 480 pixels.
- The window may be resizable, but should not include the browser toolbar, menus, or location bar.
- The window must not include the masthead, footer, navigation, or headings.
- The window must include a clear way to close it. This generally means a link or button within the page content (the standard [x] button for the window is not sufficient).
- As with all pop-up windows, the content page should not be secured as this will cause a prompt to the user to log in again.

Alt and Title Attributes

In addition to the common attributes needed to show a graphic, the following rules for 'tool tips' and alternate text should be observed to enhance usability:

- An alt attribute is required and should describe the picture, such as "LM power plant picture." If the image is also being used as a link, then the text of alt must also include a brief description of where the link takes the user, such as "plant picture, jump to product page."

> **Note:** This is a GE corporate-wide standard to ensure that web pages do not discriminate against users with visual impairments.

- If the image is also being used as a link, a title attribute is required and should indicate where the link takes the user, such as "jump to product page."
- The text for either attribute must follow common grammar and spelling rules.

Common Interaction Standards

What is "interaction?"

"Interaction" refers to navigation and any other type of interaction for users. "Interactive element" refers to the actual page elements that perform an interaction.

Basic Principles

Four basic principles should be applied to all interactive elements:

- **Distinctiveness:** the visual design of different types of elements must be recognizably distinct to assist users in finding and recognizing interactive elements. For example, using a specific color to identify 'clickable' elements or using a triangle after the menu items that have submenus.
- **Consistency:** must be applied to the following characteristics of functional interaction components:
  - Naming or labeling of components.
  - Graphical design such as color, size, shape, and the placement of elements on the page.
  - Interactive design attributes such as roll-over effects, active highlighting, and depressed/released (un-depressed) dimensional effects.
- **Alignment/Proximity:** visual elements in a page should be placed to maximize their relationships to other elements with functional, business, or content similarities. Align and put elements in close proximity with a common thread.
- **Continuity/Repetition:** keep interactive elements clearly related to the action or destination they lead to.

For example, do page titles match the menu items, buttons, or links that lead to them? If the menu item is "Gas and Oil" but the title of the page does not reflect either of these subjects, the continuity of user expectations has been broken even if the actual page is the correct destination.

Repetition between pages is encouraged, such as repeating a button on several pages in the same place with the same label (this can be considered continuity). Repetition within one page is discouraged to prevent user confusion. If needed, make repeated elements distinct, such as a menu item and a link on the page both with the name 'Home.'

Interaction may occur randomly and singly, page-by-page, or it may be repeated within any level of local or global scope. The following terms are used throughout the standard to help clarify usage within specific scopes:

Scope Terms

Global

Interactive elements that must be available throughout Inside GEPS and all associated web sites and applications.

> **Note:** Applications may remove some global interaction where it interferes with security, sessions, functional integrity, or data integrity requirements of the applications.

Regional

Interactive elements that must be consistently available throughout one web site or application.

Local

Interactive elements that are repeated as appropriate to meet functional or usability goals within a web site or application.

Independent

Interactive elements that are unique to a page of a web site or application.

Common Label Naming Standards

Labels are the text used to visually identify GUI elements that users may interact with, such as menus, buttons, fields and other form elements, or graphics used as links. Links are not treated as GUI elements for this standard, as the text of the links is itself a label and frequently must fit within the context of a sentence.

Wording

Use labels that start with a verb for GUI elements that precipitate an action. The preferred wording is **'verb object'**. Some additional guidelines include:

- Choose verbs specific to the action, for example, 'Order' is more specific than 'Submit.'
- **Always** use verbs consistently to indicate one action. For example, using 'Cancel' on a button that returns the user to the previous page and also 'Cancel Order' on a button that creates a cancellation transaction is confusing and inconsistent.
- Omit the object when it is obvious, such as the object of 'Print' a button.
- Omit the verb if it is either obvious or present from context. For example, an 'Options' menu doesn't have to be 'Set Options.'
- Omit verbs when nouns are commonly used or more intuitive. For example, a 'Product' menu commonly lists products by proper names. Or, a 'Tools' button that leads to a variety of utilities that a user may need.
- Short but specific labels are best for GUI elements with limited space, such as menus and buttons. In these cases a limit of 1-3 words is optimum. Clarity and ease of use, of course, may take precedence.

Capitalization

Labels should use title case, also known as initial capitalization (or initial caps). This capitalizes the first letter of every word within a label with the following exceptions:

- Articles (a, an, and the) are not capitalized unless they are the first word in the label.
- Prepositions (such as of, for, and by) are not capitalized.
- Conjunctions (such as and, or, and but) are not capitalized.
- 'e' used as a prefix to indicate 'electronic' is not capitalized and is followed by a hyphen. For example e-business, not Ebusiness.

> **Note:** The capitalization and hyphenation rules for 'e' used to indicate 'electronic' do not apply to logos or proper names of products. Although these standards should be considered carefully in naming features or applications, the marketing or business reasons for ignoring this rule may take precedence. Because logos generally are not used within sentences or other content -- they are not 'read' within a larger context -- the hyphenation rule is not critical to users' understanding.

- Abbreviations and acronyms should be fully capitalized. Proper nouns should be capitalized as the name requires, especially for product or business names.

Length

Length restrictions are specific to both the category of GUI element and the specific design and implementation limitations. See the sections for each type of GUI element for specific standards.

The following guidelines, however, are generally true:

- For GUI elements with limited space, such as menu items or button names, limit the length to 35 characters or less. Labels for space-restricted GUI elements should not exceed 50 characters.
- Labels should not wrap into two lines within the space allowed. This makes it much more difficult to read and can also interfere with clear distinctions in GUI elements that list items, such as menus.

Simplifying Labels to Fit Space Constraints

Space constraints for a label may require using several techniques to allow it to fit. Use these techniques, in preferred order:

1. Simplify the label -- rewrite it with smaller or simpler words. This can frequently be done without significant loss of clarity or ease of understanding for users.
2. Remove unnecessary words such as articles (a, an, the) or prepositions (of, by, for, and so on).
3. Use the following punctuation to replace conjunctions:

   & replaces 'and' - be aware that this character **must be represented by its character entity** with XHTML as it is also an XML delimiter. This does not affect the display of the character, simply the technical implementation.

   / replaces 'or' - for example 'new/used' for 'new or used'.
4. Abbreviate words carefully. Do not make them too ambiguous or general.
5. Hyphenate only if absolutely necessary.

Abbreviation

Do not use abbreviations in the labels or text associated with GUI elements. There are only two situations where exceptions to this standard are permitted:

- Industry standards for a term use the abbreviation rather than the full, proper name. An example is HTML, which stands for the Hyper-Text Markup Language. The abbreviation is more easily understood and intuitive, and therefore used more often.

- Space considerations don't permit. This should only be used after all other possible solutions have been tried. If an abbreviation is required, follow these standards:
- Be consistent!
- Choose industry standard or very well known abbreviations.
- Do not abbreviate proper nouns unless there is absolutely no other alternative. Be sure to get full approval from the content owner (Initiative leaders, business owners, marketing, etc.).
- Make sure that the abbreviation is understandable to the users or audience.
- Use ebiz as an abbreviation for e-business. The term 'ebus' is prohibited as an abbreviation.

Hyphenation

Do not hyphenate words within labels for GUI elements. Hyphenation within content text should follow common grammar rules.

The only exception to this rule is in situations where absolutely no further abbreviation or simplification of a label is possible and the space requirements for the label force hyphenation to allow the label to fit. Follow standard English rules for hyphenation in this case.

Use a hyphen between 'e' and other words where this indicates 'electronic.' For example: e-Business, not EBusiness.

Common Coding Standards

The need for consistent, standardized code cannot be overemphasized. This is especially important as we move towards new technology bases, such as XHTML, WML, and other XML vocabularies or standards.

With the GEPS Global UI, we are moving to meet the XHTML 1.0 standard. To meet this goal, four simple requirements must be met for all web site and application pages. See the XHTML Coding section on page 49 for specifics.

> **Note:** Detailed information regarding XHTML, XML, and XSLT can be found at the W3C web site.

To ensure that coding matches these standards, pages cannot be written with any WYSIWYG tools, such as Microsoft Frontpage or Macromedia Dreamweaver. These editors do not validate the code and frequently contain nonstandard additions.

General Conventions

In addition to the standards presented in this section, review the Platforms section on page 37 and Design Criteria section on page 40 for specific information, such as browser support or window width, that affects coding requirements.

- **Comments:** each page should contain a comment identifying the developer's e-mail address and the author (business group) the page was developed for, usually inside the head element. This area should also have a comment identifying the initial creation date and a modification history for the page (who, when, what, and why).
- **White Space and Other Code Practices:**
  - Use indentation consistently for easy readability. This is especially important with nested tables.
  - Remove all extra blank lines that are not required to allow easy readability. No more than one blank line is needed to separate lines or sections of code. Keeping code as short as it can be makes it easier to read and update.
  - Do not use tabs based on space characters.
  - Break elements (br) should come at the end of a line of code and before the space break.
- **Nested Tables:** use proper indentation. If the page requires more than three nested tables, consider a DHTML solution instead. If more nesting is required, no more than ten tables may be nested within a page.
- **URLs and Paths:** only relative path names are allowed. Do not use absolute paths without a domain name (i.e., no paths that assume the doc-root of the web server such as /img/myimage.gif). Absolute URLs with domain names are allowed. Relative paths that traverse up the directory tree (with ../) cannot move out of the web site or application directory tree.
- **Site Organization:** shared resources that are used repeatedly within each web site or application should be placed in a central area relative to the web site or application. Consider JSP as a way to easily include shared content for simple reuse.
- **User Alerts:** a JavaScript Confirm box is required for any interaction that leaves the current page and results in loss of data. This allows users to recover from choices with serious consequences.

XHTML Coding

Forbidden Elements

To ensure global 'look and feel' requirements throughout Inside GEPS and its subsidiary web sites and applications, the following XHTML elements and attributes must not be used:

- b or strong elements: instead, use a span or div element with a class value of "bold".
- i or em elements: avoid italics within a page to help ensure readability.
- blockquote element: instead, use adiv element with a classvalue of "blockquote" or provide local styles that handle your indentation needs.
- hr element: instead, use a repeating image (usually just one pixel wide) within a table cell to generate horizontal or vertical lines.
- font element: instead, use the standard cascading styles.
- style attribute: instead, use the standard cascading styles or additional styles specific to your web site or application. With table elements, and their children, you may also use attributes such as bgcolor, align, and so on.

> **Note:** One exception is allowed to this rule as a work-around for an IE 5.0 bug. In IE 5.0+, many CSS properties relating to absolute positioning are not handled properly when set from styles defined within external stylesheets or even styles defined with the style element in a page.
>
> To get around this, you may use the style attribute on page elements to set CSS properties for absolute positioning such as z-index, top, or left.

See the CSS Coding section on page 51 and Chapter 4 on page 59 for more information on how to use cascading styles to replace font or style in your code.

Document Type and Head Content

A document type declaration is required at the beginning of each page. Currently, the HTML 4.01 declaration should be used because of a defect in Netscape 6.0. Eventually, both the XML declaration and XHTML document type declaration will be required. Both the HTML and XHTML document type declarations are shown below (**note:** the system path in the declaration can be a local system path or a URL to the W3C web site).

Example 8. HTML and XHTML Doctype Declarations

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 strict//EN">
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

The meta element is required and should contain a description, the name of the author, and the name of the business person who owns the content.

A title element is always required.

Well-formed Code

All code must be well-formed according to the XML standard. This simply requires that the page is fully enclosed in one element (html) and every element with content (other elements or text) must have both a start-tag and an end-tag. For example, must have a matching .

Lower-case Names

Because XML is case-sensitive, the XHTML standard requires lowercase element and attribute names. So, <HTML> is not valid, but <html> is. In many cases, editing tools can automatically change the case of code for you.

Quoted Attribute Values

All attribute values must be enclosed in quotation marks. So width=50is not valid, but width="50" is. Editing tools can also help ensure that this happens.

Empty Element Syntax

Empty elements, such as img or br, do not contain other elements or text. The XML syntax for an empty tag adds a slash. So, <img src="myfile.gif"> is not valid, but is.

The 4.x browsers can have problems with this syntax if the element has no attributes. For example
 may not be properly recognized. The work-around is simply to add a space such as
.

Frames

Frames are not allowed. Use DHTML instead. In rare cases, approved technology platforms require frames.

If there is a functional need to show data continuously and a DHTML solution does not work, then frames can be used. However, only three scrollable or nested frames are allowed for the entire application. See the

Using Frames in Primary Pages section on page 26 for implementation information.

CSS Coding

To ensure that all web sites and applications within Inside GEPS have a shared 'look' that reinforces the integration and 'single portal' view, web sites and applications must use global styles for text and follow specific rules on how to apply styles to page elements.

Forbidden Style Components

Specific XHTML elements are prohibited (see the XHTML Coding section on page 49). In addition, the style attribute is prohibited except as noted in XHTML Coding section on page 49.

Instead, styles should be applied to text or table elements using the class or id attributes and a named CSS style.

Global Styles

This section discusses the rules about when to use the global stylesheet and also lists the styles within the global stylesheet that may be applied to any element on the page. Styles that apply to specific XHTML elements are discussed in Chapter 4 on page 59.

All styles in the global stylesheet are defined using a name for the class attribute. The stylesheetDemo.jsp and stylesheetExample.jsp sample pages from the /examples/intranet directory illustrates many of the styles defined in the global stylesheet.

The global stylesheets provided in the GEPS Global UI provide some very basic control for text and table elements to ensure a common 'look' across the web sites and applications. The following general rules apply:

- All text within a web site or application must use styles from the global stylesheet.
- Help content must use styles from the global help stylesheet. Help authored in the GEPS XMetaL Help system use these styles automatically.
- Links within the site **must** use the global styles to ensure that link appearance is identical across all web sites and applications. See the Buttons and Links section on page 64 for information.
- The global stylesheet sets the following CSS properties. Local stylesheets should **not** set these properties unless they must be combined with other properties (such as margin or padding) to allow a specific element to fit properly within the page.

Table 4. Global or Local Rules for CSS Properties

CSS Properties	Set in Global CSS	Set in Local CSS
Font-family	Yes. You may change the serif style (such as changing to san-serif) of text using the altSerif CSS style.	If this property must be set locally, the valid list of fonts is Bookman Old Style, Georgia, Times New Roman, and serif -- for body text and Verdana, Arial, Helvetica, and san-serif for headings or titles.. **Note:** For applications that may move to the Extranet, the allowed font list is san-serif fonts only.
Font-size	ï large: one size larger than default body text ï medium: default body text size ï small: one size smaller than default body text ï tiny: smallest text size, use **only** for very short phrases, footnotes, or legal text	If font sizes must be set within local styles, the default body text size is 10pt (13px). Other valid sizes include 12pt (15px), 9pt (12px), and 8pt (10px).
Font-weight	Use class="bold"	
Font-style and Font-variant	No	Do not use.
Color	See Table 5 on page 54.	Use colors from Global CSS or from approved palettes.
Background-color	See Table 7 on page 70.	

Table 4. Global or Local Rules for CSS Properties (continued)

CSS Properties	Set in Global CSS	Set in Local CSS
background-image	No.	Can be set locally, but use carefully. Netscape 4.7 is particularly unreliable with this property. If you use background images, the content should be isolated within a separate table cell to help ensure compatibility. This is particularly true if content has elements with an href attribute.
Padding	The default padding is zero (allows masthead and footer to span the window). Some convenience styles are also supplied: ï paddingTiny (5px) ï paddingSmall (10px) ï paddingMedium (20px) ï paddingLarge (30px)	May be set locally.
Any properties for lists, alignment, spacing (such as margins, indents, padding, whitespace), borders and boxes, height and width	No	Set locally.

Local styles generally should **not** be defined within individual pages of the site or application. Instead, define them in local stylesheets. Local stylesheet files should be named for the purpose they serve, such as localLayout.css or localForms.css. Do **not** name the local stylesheet file stylesheet.css.

Table 5. CSS Styles for Text Colors

Styles for Text on White or Light Backgrounds		
CSS Class	**Color**	**Usage**
warmColor	 #886644	Use for any text element on a white or light background.
coolColor	 #003399	
coolNeutral	 #666677	
warmNeutral	 #777766	
blackText	Available on for the Intranet, this style sets the color to black. May be used with heading elements, such as h5.	

Table 5. CSS Styles for Text Colors (continued)

Styles for Text on White or Light Backgrounds		
CSS Class	**Color**	**Usage**
errorMsgStrong	#AA3333	Use to highlight specific content in error message pages.
errorMsgHilite	#999944	
Styles for Text on Dark Backgrounds		
CSS Class	**Color**	**Usage**
onDarkBgColor	White, #ffffff	Any of these colors may be used for text elements of any kind when they are used against a dark background.
onDarkBgWar-mMedium	#BBAA66	
onDarkBgWarmLight	#FFF3AA	
onDarkBgWarmBright	#FFFF00	
onDarkBgCool1	#B3C6FF	
onDarkBgCool2	#5983FF	

File Naming Conventions

These are the basic naming standards for all files used in the user interface:

- File names must be all lowercase. Do not use spaces in file names.
- Avoid hyphens, underlines or periods as word delimiters as these may not be valid name characters in many situations.
- Use full English names. They are easier to understand and require less additional documentation to explain.
- Make the names fit the subject or object. For example, neworderpage.jsp or adminmenu.shtml.
- Keep names to 15 characters or less if possible. This rule may come in conflict with the full English names rule - try to balance clear names against excessive length.
- Abbreviations should be limited to package names or to abbreviations and acronyms that are in common use within the business, such as GENE (for GE Nuclear Energy). Use abbreviations only where it is necessary to keep the length within reason.
- Avoid names that are very similar or only differ by case or plurality. For example, order.html and ordered.html should not be used together.
- Hungarian notation or other naming standards that involve abbreviations or characters to supply typing or categorization within names should **not** be used.
- Do not name a stylesheet stylesheet.css. Give the stylesheet a logical name describing what it provides.

Capitalization in Code

Common coding styles for specific elements within JavaScript, XHTML, or in other domains have different capitalization requirements. **Specific requirements for different page elements override this general standard.** For specific information, see the XHTML Coding section on page 49 and the Capitalization section on page 45.

In general, mixed case (both lower- and upper-case characters) should be used. Use an initial capital letter to highlight the beginnings of words within names rather than using other punctuation. For example:

userLogin

rather than

user_login or user-login

Comments

Comments should be simple and add clarity. They should clarify *why* an action happens -- not just *what* is happening. See also the General Conventions section on page 48 for required commenting in a page.

Determine your comments before you write code (this is useful to organize and clarify the code up front) or at least during coding. Do *not* wait until the end to add comments.

Function and variable names in JavaScript should be self-explanatory and the functions should be simple. Comments within the function must cover the following topics:

- The purpose of the function
- Author
- Version
- Date

Additional comments are allowed. If the functions are becoming complex and require more comments, it may be simpler to redesign the functions into smaller modules.

Chapter 4: Page Element Standards

Text Elements

Text elements must use the global styles defined in Global Styles section on page 51. Additional styles are not generally allowed, unless the audience or business purpose strongly requires them. In addition, keep the following points in mind:

- **Page Body:** The standard body element assigns default font values to the page and sets a standard margin for the document. Pages developed using the template require a margin of zero pixel width for the header and footer to be flush with the browser edges.
- **Titles and Headings:** The web site or application title is defined in the standard template. See the Configuring JSP Variables for Primary Pages section on page 18 for information.

  The standard XHTML elements, h1 through h6 provide three different sizes of headings. The odd numbers (h1, h3, h5) are a normal weight and the even numbers (h2, h4, h6) are bold. You may use any of the named colors (see the Global Styles section on page 51) to apply colors to any of the heading elements.
- **Paragraphs, Lists, and Blockquotes:** paragraphs use the default font and sizes and are left aligned. Or use class="p" attribute with the div element. You may use any of the standard size or color classes (see the Global Styles section on page 51) on paragraphs.

  You may also use the altSerif style to switch the font families from serif to sans-serif for any paragraph, list, or blockquote.

  The basic styles for the following elements use the default fonts and sizes: ul, ol, dl, dt, and dd. You may use any of the standard size or color classes (see the Global Styles section on page 51) on lists. Use local styles to change the alignment, markers, padding, or margins.

  The blockquote element is forbidden. However, you can get an indented effect (both right and left margins) on a div element using the class="blockquote" attribute.
- **Inline Text Elements:** You may use the span element combined with the size, color, or weight classes from the global stylesheet to apply specific colors, sizes, or bolding to short phrases or labels (see the Global Styles section on page 51).

  You may also use the altSerif style to switch the font families from serif to sans-serif for short text phrases.

In addition, both the sup and sub elements have their font size set to ensure that superscripts or subscripts are legible.

Information and Error Messages

Validation should be completed on the client side **whenever possible**. This captures errors sooner, allowing users to correct them more quickly with less irritation. Client-side validation should occur when the user submits the page, rather than at individual fields. See also the Form Elements section on page 67 for more information on validation requirements.

All error messages need to inform the user, in a clear and concise manner, what problem caused the error. In some cases, they also need to provide additional technical information to allow support staff to determine the cause and resolution for an error.

Error messages:

- Must use complete sentences with correct spelling, grammar, and punctuation.
- Must focus on what the **user** needs to understand. To help ensure this, messages should be written or reviewed by technical communicators, analysts, or other business project members.
- Should include technical explanations, jargon or system identifiers (error numbers or codes) **only** if:
  - This information is provided in addition to user-level explanations.
  - The error relates to system problems. This detail can help support staff identify the problem and needed resolution.
  - Explanations of system identifiers and their technical implications are made available to support staff.
- Must use a JavaScript alert box with an **OK** button for all client-side error messages.
- Should aggregate multiple, client-side errors into one message. Consider listing errors or using a general message identifying errors by some highlighting technique (such as all fields in red with white text).
- Must use an error message page for server-side errors. This page must offer one of the following resolution paths to the user depending on application needs and the cause of the error:
  - The user can end the use case but continue working in the application.
  - The user can return to the use case where the error took place to fix the error and continue the use case.
  - The application closes after the user acknowledges the error. In this case, the error message page **must** contain a warning to inform the user that the application will terminate and indicate any implications of that termination.

– For session time-out errors, the user may log in again. In this case, the error message page must clearly indicate that incomplete transactions or data from the expired session may be lost.

In addition, the error message page must include the following information for system-related errors:

– The Help Desk (application support) phone number **must** appear on the error message page to encourage users to notify support staff.
– System identifiers or more technical information should appear to assist support staff. This is **strongly** encouraged.

CSS Styles for Error Messages

Client-side error message layout and styles do not permit a significant level of control. Server-side errors, however, are presented in a new page and thus may use any of the global styles available for content (see the Global Styles section on page 51). In addition, two specific styles are defined in the global stylesheet to highlight information as needed, errorMsgStrong and errorMsgHilite

Interactive Elements

Categories

The primary categories of interactive elements include:

- **Menus**: menus are a grouping of labels (either graphical or text) that provide access to an organized hierarchy of options. Menus are frequently oriented in horizontal or vertical bands. The various menu options also typically appear in 'pull-down' lists.

  This document uses the term 'pull-down' to indicate access to the next level of menu or branching. This does **not** imply a required implementation. The actual visual implementation to present menu options may use other methods of displaying choices.

  > **Note:** See the GEPS Intranet Menu Standards for the standards and implementation instructions for menus in the GEPS Global UI.

- **Tabs**: tabs are a grouping of labels (either graphical or text) that specifically look like a 'file folder tab' using a curved extrusion with a label as the interactive area of functionality.

  Tabs provide access to individual options, but no hierarchy is involved. Also tabs typically remain on a page with only one active and visibly 'in front.' Choosing another tab moves the information associated with that option to the front and information from the previous tab is hidden.

- **Toolbars**: toolbars are organizations of buttons or links, typically with a shaded bar behind the buttons to reinforce the organization, although other organization methods may be used. Toolbars provide access to individual options, but no hierarchy is involved. Multiple toolbars may be used to categorize buttons or links.
- **Buttons**: buttons are textual or graphical labels, typically with a background color to help make the button more visible. Buttons may appear 'flat' or have a three-dimensional appearance to reinforce the 'pressed/depressed' paradigm users are accustomed to with buttons in physical devices. Buttons provide access to individual options that may or may not be related to any other interactive element on the page.
- **Links**: links are textual labels, typically displayed in a specific color with other visual hinting mechanisms to help users recognize the label as an interactive element. Links may occur separately in organized areas or lists -- or they may occur within the flow of text on a page. Like buttons, links provide access to individual options that may or may not be related to any other interactive element on the page.

Navigation Standards

Menus are the preferred paradigm (the interactive category) for navigation and functional flow in web sites and applications, where the design criteria for menus meet the requirements of use cases and the analysis of user profile needs. Applications may use other paradigms for functional flow, if menus do not meet these requirements.

See the *GEPS Menu Standards (XML Menuing System)* for information on menu standards and instructions on configuring menus with the XML Menu System.

Tabs

Tab Design Criteria

Tabs are somewhat similar to menus in that they organize choices and keep choices available within the current scope (usually regional or local). However, no deep hierarchical organization is possible. The most common reasons to use tabs are for applications with highly-related data entry or display use cases.

Tabs should only be used when the following criteria are true:

- There is only one level, or at most two levels of organization for the choices.

  An example of a two-level organization for tabs would be data entry pages for a sender and recipient where the user may need to supply address information for both and billing information for one or the other. This could be done as a single level Sender Address, Sender

Billing, Recipient Address, and Recipient Billing) or as two levels (Sender and Recipient vertically and Address and Billing horizontally).

- The scope of the choices is at least local or regional and it is important, either functionally or for usability reasons, that access to the choices be kept together and always available within the current scope.
- Choices are independent -- there is no prescribed order or chronology that users must complete choices in.

> **Note:** It is acceptable that data from one tab either defaults or determines data in another tab as long as the related data in the second tab is either optional or can be completed by users independently of completion of the first tab.

- Choices are highly related to each other in some logical or functional order.
- Only one choice can be active (currently displayed) at a time.

The visual design of tabs must match user expectations and be organized horizontally (preferred), vertically (less preferred), or both (for two level organizations only). They must also recognizably look like 'folders tabs' using a curved extrusion with a label as the interactive area of functionality.

Roll-overs

Roll-overs may be used in tabs, but must be used consistently and **sparingly**. Too much highlighting becomes ineffective.

Tab Naming Standards

Tab labels should follow the common naming standards, especially those relating to space-limited GUI elements. See the Common Label Naming Standards section on page 44. When space is particularly tight, tab labels may wrap to two lines.

Toolbars

Use toolbars to organize buttons or links into categories. They are frequently used to allow access to a set of buttons across a local, regional, or global scope. The criteria to choose toolbars is:

- There is an organization to the choices, but only a single hierarchical level. Choices may be organized into different categories.
- The scope of the choices is local, regional, or global.
- Choices are independent. There is **no** prescribed order or chronology that users must complete choices in.

Toolbar Design Criteria

Toolbars may use color, borders, or simple proximity and white space to define their organization. They may contain buttons or links, but generally shouldn't include both within the same toolbar (all buttons or all links).

The organization of buttons or links within the toolbar should be based on the functional needs of users. This can be alphabetical, organized by frequency of use, related functions, or any other principle that fits the users needs.

Buttons or links within a toolbar should be similar sizes, wherever possible, to help emphasize the relationship within the toolbar. This is usually more practical with buttons, where icons can be used.

Toolbar Naming Standards

Toolbars are not required to be labeled. If a label is used, it should follow all the common naming standards. See the Common Label Naming Standards section on page 44.

Buttons and links within the toolbar should follow their own naming standards, although buttons are more typically labeled graphically with an icon.

Buttons and Links

Button or Link Design Criteria

Buttons and links serve the same basic purpose; they are simply different visual implementations of the same functional interaction. Whether an interactive element should be a link or a button is sometimes unclear, so the standards discuss them jointly.

Because of the different visual implementations, however, they are frequently used in slightly different situations. Buttons are usually more visually prominent and thus used more often for discrete and significant functional choices that do not require contiguous text to explain. Links may be used for less prominent functional choices or in any situation where contiguous text explaining the context of the choice is appropriate.

Buttons and links can also be organized into Toolbars section on page 63. Buttons and links should be used in the following situations:

- The scope of the interaction may be independent (used only on one page) or on any repeated level (local, regional, or global).
- The interaction has a single level (no hierarchy).
- Choices of links/buttons on any one page are independent. The user has full discretion of choice.

> **Note:** This independence does not necessarily imply a lack of relationship with other choices on the page. Buttons and links can be grouped either in simple spacial organizations or in toolbars.

- Choices may have a prescribed order or chronology through a use case, although this is not required.

  For example, a use case that has a required order may use **Back** and **Next** buttons to provide access through the ordered pages in a single direction. The buttons are independent -- users can choose either, but the use case has a required order to be complete. Users cannot exit this prescribed order except through other buttons (such as **Exit**) that properly handle that choice.

Button or Link Labels

Button labels should follow the common naming standards, especially those related to space-limited GUI elements. See the Common Label Naming Standards section on page 44. Additional special considerations for buttons include:

- Buttons that are grouped together should be the same size to create a balanced design. In these cases, the length of the label should be consistent.

  Parallelism in wording is also a good rule to follow, where possible, with grouped buttons. For example, if one button in a group starts with a verb, all buttons in the group should start with a verb.
- Button labels generally should not wrap to two lines, but this prohibition is not as strong as for menu items. If wrapping is used, to fit within space constraints for example, it should be used consistently with that button and with any buttons that are grouped with that button.
- Buttons that appear on multiple pages must use the **exact same label** if they perform the same action.

In addition, the following standard button labels are required:

Table 6. Required Button Labels

Button Label	Action
Exit	A button that provides an escape mechanism to leave an application and return to a web site. This should not be used for the standard method of exiting the application (if there is one), but specifically for an 'escape hatch.' The button is used to properly close any outstanding session, state, or security issues.
Next	In functions that have an order in which actions must be performed, this button allows the user to continue to the next page or action. It should be used in conjunction with the Back button. If a more specific verb exists for this action, such as at the end of the function when a transaction is submitted or some data is saved, that verb may be used on the page where that action occurs.
Back	In functions that have an order in which actions must be performed, this button allows the user to return to the previous page or action. It should be used in conjunction with the Next button. This button may be omitted when the use case does not allow the user to return to a previous action. This is typically true in situations where a transaction has already been submitted and the user should be forced to submit an update or cancellation rather than simply returning and changing content.
Finish	A button that concludes the normal use of an application, if such exists, and returns the user to a web site. The button is used to properly close any outstanding session, state, or security issue. Not all applications have a defined completion point, but if one exists, this button should be used. If a more specific verb exists for the completion point within the application, that verb may be used as the button label. This is provided as a generic wording.

Button or Link CSS Styles

All links use the global styles with no class attribute to ensure that links are consistent through all web sites and applications. Link styles are currently defined for the :link, :visited, :active, and :hover pseudo properties.

> **Note:** The standard link colors and decorations work on white or medium backgrounds only. Do **not** place links on dark backgrounds.

Button or Link Coding

Links (a elements) must always have a title attribute to display a 'tool tip' indicating the target of the link. Links must also have an onMouseOver attribute that updates the status bar with the name of the target for the link.

Form Elements

General Conventions

- Form elements require unique names with no underscoring.
- Forms should be validated on the client using JavaScript unless validation from the server is absolutely mandatory.
- Validation should occur when the entire form is submitted by the user. Field-by-field validation is not recommended unless input of a later field is dependent on the value of the current field, or a repetitious error would cause a large amount of rework for the user.
- Upon submission, the form remains populated with any erroneous data appearing in white font upon a red background. (The color choices are recommended for consistency; however, if the colors are likely to deeply offend readers of certain cultures then modifications are allowed.)
- See also the Information and Error Messages section on page 60 for more information on validation requirements and the resulting error messages.

Fields

Field Labels

Field labels should follow the common naming standards, but space is typically less constrained. Therefore, standards for space-limited GUI elements may not apply, and exceptions due to space constraints may not apply to field labels in all cases. See the Common Label Naming Standards section on page 44.

Specific areas where field labels may follow different standards include:

- **Wording:** since fields do not typically represent a specific action, field labels generally should be nouns rather than verbs unless the field actually represents an action to be taken. Address, for example, is a legitimate field label. Print Request may also be a legitimate label for a check box indicating that the user wants a hard copy of the current request once it is saved.
- **Sentences:** Field labels may consist of complete sentences, although this should be used judiciously. For example, a sentence explaining the implications of an option for a check box is an appropriate use for a field label. A sentence describing one line of an address is not appropriate.
- **Shortening Labels:** limiting words is not as important as clarity. Techniques used to shorten labels generally should not be used for field labels.
- **Table Headings:** column and row headings where data is displayed in tables may fall into the space-limited category. In rare cases, space can be so limited that even good abbreviation techniques are not sufficient. If no other design options are possible, column and row headings may be as small as a single character when space does not permit otherwise.

Field labels should also be placed to the left of the actual field, unless fields appear in a table layout where field labels may be used as column headers instead. The justification and spacing between the label and the field depends upon the page real estate, but at least one space should separate the label from the field.

Selection and Sorting Elements

Several types of elements can be used to allow users to select options, values, or specific records (data with more than a single value that is logically related) or to sort records.

Selection Techniques

- **Radio buttons**: allow the user to select one option, with a single value, from a short list (usually < 12). Radio buttons are ideal for questions with a yes/no or true/false answer. They are not appropriate for selections that require multiple values (records) or with a medium to large number of values.
- **Check boxes**: enable the user to select all, some, or none of a group of single-valued options. Groups of check boxes work well in situations when the user is not familiar with all of the possible options, where options have a yes/no or true/false answer, or where users must be able to select multiple options. Check boxes are inappropriate when

the selection requires multiple values (records) or there are a medium to large number of values (generally > 20).

- **Selection boxes**: enable the user to select one or several values from a limited number of single-value options. Selection boxes are also known as 'pull-down lists' or 'pick lists.'

  Selection boxes are ideal when options are familiar and there are more than a few (usually > 6 and <75). For example, choosing a US State for an address. Selection boxes are not appropriate when the determining the selection requires display of more than a single value (record selection) or when the number of choices is large (> 75) or unknown.

- **Links or Buttons**: use links or buttons to allow users to select a record (multiple values) and perform an action on that record. One common example is to make the data in one column of a list of records into links that allow the user to select a single record and then perform an action (update it, display details, etc.)

  Links or buttons should be used as the **preferred method for record selection** and action as they allow users to select and perform the action with a single click. All other methods require two clicks (select using something like a radio button and then act using another link or button).

Sorting Techniques

There are several ways to allow users to sort records on a page. In general, sorting techniques should use the smallest number of clicks needed to accomplish the functionality.

One preferred method to allow sorting is to use links or buttons as column labels for the fields within the records that allow sorting. Users click the column heading (a single click) to sort by that column.

A less preferred method would use some type of selection technique to identify the sorting criteria, such as radio buttons, and then a separate button to perform the sort. This mechanism may be needed if sorting involves more than a single level.

Tables

The table element may be used for page layout or for content (or both). Layout properties, such as alignment, margins, padding, borders, and so on, are page- and site-specific and should be defined in local stylesheets.

Note: The global stylesheet does not, with a few specific exceptions, define values for margins, padding, alignment, or other layout/white-space properties. However, content almost always needs some control to display pages correctly. This is especially true for multi-column layouts that are compatible across all supported browsers.

For example, Netscape 4.7 frequently does not apply column widths properly in pages with multiple columns when the columns are inside a table with width="100%". Using a table element with a 100% width is frequently used to ensure that the page content scales nicely as users resize the window, but this creates compatibility problems.

The fix to this problem is to set both the margin and padding properties on table cells for content.

To ensure that body text is applied appropriately in all browsers, the global stylesheet also defines the basic fonts and font size on td (table cell). It also defines a style for th (column or row header cells) that bolds the text. You may also use the padding, color, and size styles from the global stylesheet (see the Global Styles section on page 51).

In addition, the global stylesheet defines several styles with background colors for use in table, tr (rows), and td (cells). These styles may be used to set up tables with alternating rows of colors that work well with the color palettes. This alternating color layout is fairly common in some types of applications, but any of the standard background colors may be used to provide highlighted areas.

Table 7. CSS Styles for Background Colors on Tables, Rows, or Cells

Class	Color
tableHeadCoolNeutral	#CCCCDD **Note:** This class also makes any text bold.
rowCoolNeutralLight	#EEEEFA
rowCoolNeutralMedium	#D4D4EE

Table 7. CSS Styles for Background Colors on Tables, Rows, or Cells (continued)

Class	Color
tableHeadWarmNeutral	 #CCCBCB **Note:** This class also makes any text bold.
rowWarmNeutralLight	 #F8F8EB
rowWarmNeu-tralMedium	 #E6E6DA
tableHeadWarmColor	 #DDDDBB **Note:** This class also makes any text bold.
rowWarmLight	 #FFFFDD
tableHeadCool1Color	 #E7EBFF **Note:** This class also makes any text bold.
rowCool1Light	 #E7EBFF

Table 7. CSS Styles for Background Colors on Tables, Rows, or Cells (continued)

Class	Color
tableHeadCool2Color	 #B4B4FE **Note:** This class also makes any text bold.
rowCool2Light	 #D4D4FF
wizardBgColor	This is only available on table and should only be used for 'wizard-type' applications.  #BBBBBB

Pop-up Windows

Pop-up windows open another browser session from the primary browser and are useful for presenting secondary information, such as online help or legal agreements. Because they create separate browser sessions, they are also useful to isolate applications entirely or within applications to minimize session or state impacts from links that lead outside the application.

The following general rules apply to pop-up windows:

- Web sites may use the standard pop-up window for online help. Avoid pop-up windows for web sites, unless no other option provides the required functionality.
- Applications **must** use the standard pop-up window provided in the Global UI for online help.

  Other types of content can use pop-up windows (the standard one provided in the Global UI or any other) to display content such as:

  – Online help (icon in standard masthead)
  – Site map (icon in the standard masthead)
  – Tutorials (application-specific)
  – Oversized graphics (see the Oversized Graphics section on page 42)

- Applications should carefully consider pushing any type of secured content to a pop-up window. Because the new window creates a new session, user authentication information no longer applies and users will be prompted to log in again if the content is secured.
- Pop-up windows should be used **sparingly** as they are very distracting to users and can be very confusing. In general, no more than three different pop-up windows should be used -- **including** the standard pop-up window.

> **Note:** The standard pop-up window is designed so that only one instance of the window may be open. If users open the window for one type of content, such as a site map, leave the window open, and then request some other secondary content that uses the standard pop-up window, the new request **will replace** the previous content in the standard pop-up window.

- Unless required for functional reasons, non-standard application pop-up windows should open initially at a smaller size than the primary browser minimum (smaller than 800 x 600 pixels) and should be resizable. As they present secondary content, pop-up windows should not completely obscure the primary browser window.

  The standard pop-up window opens at a default size of 450 x 500 pixels and is resizable by the user.
- Other functionality of non-standard pop-up windows, such as the ability to resize the window or the presence of the browser toolbar, should be dictated by the functional requirements of the secondary content.

Pop-up Window Coding Standards

See the Configuring JSP Variables for Pop-up Pages section on page 33 for an example of the template for the standard pop-up window. All other code should follow the general coding standards for XHTML, JavaScript, DHTML, and so on.

Pop-up Window CSS Styles

Pop-up windows should use the global CSS styles for text just as they would in primary content pages. This is automatically linked into the standard pop-up window. Specialized style requirements can be added to local stylesheets. Tutorials, for example, frequently have specific needs that may not be addressed by the global styles.

Online help **must** use the specialized global styles for help. With the standard pop-up window, these styles are automatically linked.

Online help that is authored with the GEPS XML Help application (in XMetaL) uses the global CSS styles automatically. Help systems that are authored directly in HTML **must** use the global help CSS styles. See the Online Help Standards document for more information on the GEPS XML Help application and online help standards.

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.

client

In the client/server model of communications, the client is the machine connected to a network that remotely accesses the resources of a computer server. Client may also refer to the person using the client machine. The term client may also refer to any software that is used on a client machine to exchange data with software running on the server, such as an e-mail client.

color palette

A set of colors that have been chosen to work together in some specific context. For the GEPS Global UI, the approved color palettes have been chosen to blend well with the standard masthead and footer and to work well in graphics or text.

cookie

A mechanism used by server-side connections to store and retrieve information on the client side of a web connection. Cookies are small data files that contain information the web site can use to track things, such as passwords, pages visited, and the date when you last looked at a certain page.

CSS

Cascading Style Sheet: A standard used to define styles on a single web page or a group of web pages. The styles determine how information is displayed in browsers. For example: You could define a style for text that sets the font to Arial, and the size to 2, and use that style where it is appropriate on your HTML pages. A style can affect one page, or a group of pages. Current browsers support CSS.

DHTLM

[HTML] Dynamic HyperText Markup Language: A markup language, used by Netscape and Internet Explorer browsers, that combines HTML, stylesheets, and scripts to make Web pages more interactive. DHTML supports Cascading Style Sheets (CSS).

doc-root

A directory for a web server that serves as the root directory for all content that the server may serve. Relative paths, such as "/products/my-page.html," are relative to the

doc-root. The web server uses doc-root to supply the remaining portions of the path in order to find the page to serve.

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.

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.

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.

HTML

HyperText Markup Language: A fixed format, based on SGML, for hypertext documents on the Internet. It is very simple and allows for the embedding of images, sounds, video streams, form fields and simple text formatting. References to other objects are embedded using URLs.

JavaScript

A Web scripting language designed by Netscape to be used in HTML documents. It is used in browsers and Web servers primarily to tie components together or to accept user input. JavaScript is unrelated to Java.

JSP

[Java] JavaServer Pages: An extensible Web technology used to return dynamic content to a client. JSP uses HTML or

XML data elements, scripting languages, and server-side Java objects. In most cases the client is a Web browser.

page

[Usage] One of a collection of Web documents for a web application or help project. Each document is referenced by its own URL. Usage examples: Home page, or help topic page.

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.

portal

[Internet] A web site that serves as a resource, or common point of entry, to other sites on the web. Typical services offered by a portal site include a directory of web sites, a facility to search for other sites, and other items of interest, such as news items, weather information, e-mail access, stock quotes, telephone and map information, and a community forum. Some portals provide the user with the ability to personalize the site to meet individual needs and interests. The GEPS web site, gepower.com, is the Power Systems portal.

session

A session represents the period between when a user logs on and logs off of an application. The session includes all the request/reply interactions between the user and the application and may include one or more transactions. The session lasts until the user exits the application or browser (client application).

soft-link

Also known as a symbolic link, this is a specific type of directory entry in Unix platforms that allows a file or directory from one place within the file system to be treated as though it is present in other locations. The file system or web server interprets a soft link as being the target file or directory tree.

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.

state

The condition of a system or a Web page, including its attributes, configuration, and content. For a Web page, state is maintained only as long as the page is in the

	browser, unless a cookie or the application itself is set to retain information.
stylesheet	*[XML]* A list of specifications describing how to present a document in a particular medium. Cascading Style Sheets (CSS) and Extensible Style Language (XSL) are the stylesheets used in HTML or XML respectively.
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.
URL	*Uniform Resource Locator:* A resource identifier that describes its target by presenting a pathway for retrieving it. URL may include a protocol, a host computer, and how to find the target resource on that computer.
user interface	The parts of a software application or Web site with which the user interacts directly, such as the screen display, keyboard, mouse, and help messages. The user interface also refers to the appearance, response time, and content that is part of the user's experience with the application or Web site.
WAR	*[Java] Web ARchive*: A JAR with a defined directory structure to allow the archive and deployment of a web application. Supported by Servlet v. 2.2.
W3C	*World Wide Web Consortium:* An industry group created to design and promote standards to increase the functionality of the Web. The W3C was initially established in collaboration with CERN, the creators of the World Wide Web. W3C sets the standards for XML, HTML, HTTP, XSL, CSS, RDF, and other Web-oriented standards.
web-application-root	For an application server, this is a root directory from which all content for one application stems. Unlike doc-root, an application server may have many web-application-root directories -- one for each application within the server.

Relative paths, such as "/html/splash-page.jsp," are relative to the web-application-root. This is particulary important for applications that are deployed using WAR files as the J2EE standard requires that all resources relative to the application reside under the web-application root.

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.

well-formed *[XML]* A document that meets the basic document syntax rules of XML, but is not associated with a specific DTD. To be a well-formed XML document, an XML declaration must appear at the beginning of the document, a single root element must wrap the document content, elements must be nested properly, and tags must be constructed according to general rules of XML syntax.

WYSIWYG *What you see is what you get:* Describes software that allows you to perform edits on a document that show you on screen exactly what will print out. Popular in desktop publishing.

XHTML eXtensible Hyper-Text Markup Language. This is an XML-compliant version of HTML 4.0.

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

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X