

SCOPE: This standard applies to web applications (excluding SharePoint) developed or maintained by staff or consultants employed by the Office of Information Systems. Reports (output with the intent of printing) will be addressed in a future release of this standard.

STRUCTURE: Standards are listed with numerical references. (Example: Standard 1.2). Supporting Details are included after each standard. These supporting details provide some Best Practices that BSSO developers have learned over the years. It also includes references, links and techniques that can be used in conjunction with the referenced standards.

GLOSSARY: A glossary of technical terms referenced within the body of the document is available on the BSSO website (<http://infonet.dot.state.fl.us/bssso>)

STANDARD

1. Accessibility

- 1.1. All web applications must meet the standards established in Section 508 of the Rehabilitation Act.
- 1.2. For more information on Section 508 and other compliancy information you can visit <http://infonet.dot.state.fl.us/bssso/Section508.html>

SUPPORTING DETAILS - Section 1

Florida Administrative Code Rule Chapter 60EE-1 requires all State Agencies be compliant with Section 508 of the Rehabilitation Act of 1973. The requirement to follow Section 508 is a State, rather than just an OIS or BSSO requirement. A list of links related to Accessibility Information is available on the BSSO Web Site.

Reference:

BSSO Section 508 Links <http://infonet.dot.state.fl.us/bssso/Section508.html>

2. Fonts and Colors

- 2.1. All text other than error and warning messages must use the default hexadecimal font color of black. (“#000000”)
- 2.2. All error and warning messages (except for modal dialog boxes) must use the red (#FF0000) font color and be displayed on a white (#FFFFFF) background.
 - 2.2.1. The term **ERROR**: must precede the error message and render the font format of bold, red (#FF0000), and all caps.
 - 2.2.2. The term **WARNING**: must precede the warning message and render the font format of bold, red (#FF0000), and all caps.
- 2.3. The font face must use the true type font family of “Arial, Helvetica, Verdana” in the listed order. Unless monospace fonts are needed for alignment purposes, then the font family of “Courier New, Courier, Monospace” must be used.
- 2.4. Font size must fall within the ranges of Size 1 (8 point) and Size 5 (18 point).
- 2.5. Underline must not be used for anything other than hyperlinks.
- 2.6. Fonts must not blink.
- 2.7. Italics must not be used.

SUPPORTING DETAILS – Section 2.1

Using the standard Fonts and Colors ensures that all applications share a consistent or common look and feel.

SUPPORTING DETAILS – Section 2.2

Providing a standard format for Error and Warning messages will assist the user with identifying issues that arise

during use of the application. Using red font alone would be a violation of [Section 508 Subpart B, §1194.22\(c\)](#), as you would be using color coding as the sole means of conveying information. For this reason we require the message be preceded with either the term Error or Warning. An example that shows the proper formatting for an error and warning message can be seen below:

ERROR: The City and State fields have not been completed. City and State must be completed before continuing
WARNING: The City and State fields have not been completed. Your application will be processed faster if City and State are completed

A **Modal Dialog Box** (also referred to as a **Modal Window**) is a child window that requires the user to interact with it before they can return to operating the parent application, thus preventing the workflow on the application main window. Modal dialog boxes are commonly used in GUI systems to command user awareness and to display emergency states. Examples include JavaScript Alert, Prompt and Confirm boxes and AJAX Modal Popup Windows.

Generic Error Pages are expected to follow this standard.

SUPPORTING DETAILS – Section 2.3

Font Family defines the fonts that the text should be displayed in. If for some reason the Arial font is not available on the user's PC, the next available font in the family will be used to display the text. Using a set Font Family provides consistency across each file of the application. The Arial, Helvetica, Verdana fonts are most commonly found on people's computers, and are easy to read both in the browser and in print.

The Courier New, Courier, Monospace fonts may be used when alignment of text is needed. A monospace font displays like a typewriter font meaning that all characters use the same width.

SUPPORTING DETAILS – Section 2.4

Limiting the font size range provides consistency across the various applications. Fonts that are smaller than the listed limit can be hard to read to users with your most basic vision problems. Fonts larger than the listed limit take up more room than needed to convey the message. The fonts should be used consistently throughout your application. (i.e., the contact information found in the footer of each page should use the same font size(s).) Remember to keep it consistent.

SUPPORTING DETAILS – Section 2.5

Users commonly associate underlined text with a hyperlink. Restricting the use of underline to hyperlinks ensures that hyperlinks are easily recognizable and plain text is not mistaken for a hyperlink.

SUPPORTING DETAILS – Section 2.6

Not all browsers support the html <blink> tag. Techniques from "WAI Guidelines: Page Authoring" provided by the W3C provide the following information:

Avoid blinking: [Technique A.7.3](#) Authors should avoid creating motion and blinking in a page where possible; blinking may cause seizures in some users and is annoying to many other users. They should also provide a mechanism for freezing motion. If style sheets are used to create an effect (e.g., 'text-decoration: blink'), users may cancel the effect through style sheets as well.

Note. Do not use the BLINK and MARQUEE elements. These elements are not part of any W3C specification for HTML (i.e., they are non-standard elements).

Reference: WAI Guidelines: Page Authoring <http://www.w3.org/WAI/GL/wai-gl-techniques-19980918#style>

SUPPORTING DETAILS – Section 2.7

Italicized text is not always easy to read. Emphasize text by rendering it as BOLD, or by setting the font weight to BOLD.

3. **Hyperlinks**

- 3.1. Text hyperlinks must not extend beyond the element to which it is related.
- 3.2. Hyperlinks to downloadable files must include a text description that includes the file size and file type. If the resulting file size varies because the file is created on-request, then it must be stated that the file size is unknown.
- 3.3. If a plug-in is required, access to the plug-in must be provided on that page.
- 3.4. The destination of every hyperlink must be identified with descriptive text.
- 3.5. Text hyperlinks that are not within the identified Navigational Menu must adhere to the following;
 - 3.5.1. Underline must not be disabled.
 - 3.5.2. Must use the default browser hexadecimal color settings. Link use blue "#0000FF", Active Links use red "#FF0000".
 - 3.5.3. Links that reference static electronic documents, pages, or search results must use purple "#800080" to designate when the link has been visited

SUPPORTING DETAILS – Section 3.1

Hyperlinks that extend beyond text elements look to be made in error.

Examples:

Correct: We can find many examples of the FDOT logo. An example can be seen by visiting our [Infonet Website](#).

Incorrect: We can find many examples of the FDOT logo. An example can be seen by visiting our [Infonet Website](#)

SUPPORTING DETAILS – Section 3.2

Text descriptions provide information to the customer to ensure they have the proper software to view the file, and that they know when they may be attempting to download a file that is too large for their connection speed.

Example: Please review the [FDOT Organizational Chart](#) (PDF, 224 KB)

SUPPORTING DETAILS – Section 3.3

This provides easy access to the software required to view/use the content you are providing. We should not expect our customers to have to search for required software when we already know where they can find it.

It is also a requirement of [Section 508, Subpart B, §1194.22\(m\)](#) that when a "web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l)."

Many common file types are already made available to FDOT Employees or Consultants utilizing the standard desktop configuration. For this reason, links to those plug-ins are not required for Intranet Applications.

SUPPORTING DETAILS – Section 3.4

This standard provides screen readers and other assistive technology the information needed to convey the destination of the hyperlink to the customer. Screen readers will read the words as written, and will reference any hyperlinks within the text. Hearing the words "Click here" followed by a URL is not as informative as hearing "Infonet Website" followed by a URL.

Examples:

Correct: We can find many examples of the FDOT logo. An example can be seen by visiting our [Infonet Website](#).

Incorrect: We can find many examples of the FDOT logo. [Click here](#) for an example on our Infonet Website.

SUPPORTING DETAILS – Section 3.5

Hyperlink formatting is standardized to be consistent in our applications so that hyperlinks that exist within the application content are easily recognizable. This consistency makes Web applications easier to use and understand. Those areas of the application easily identified as the Navigational Menu are not required to use the underline, or the default browser hexadecimal color settings.

Some examples of "easily identifiable" Navigational Menus include:

- a) Tabs or tabular navigation
- b) Breadcrumbs
- c) Tree view
- d) Consistent location and appearance

The visited link color of purple is most useful when referencing static documents, pages and search results. The use of the visited link color is not required when a JavaScript function or link button is used because their purpose is primarily for initiating an action within the application.

Examples of these Navigational Menus can be seen in the Enterprise Library Message Board, as well as in the Glossary.

4. Performance and Page Width

- 4.1. All requested content must be received by the browser within 10 seconds of the user action.
- 4.2. All requested content received as a result of a user action must not exceed 150,000 bytes, excluding the following:
 - 4.2.1. WebResource.axd files
 - 4.2.2. Cascading Style Sheets
 - 4.2.3. .JS Files implemented as Include Files
- 4.3. All web pages must be viewable with no horizontal scrolling on a screen width of 1024 pixels.

SUPPORTING DETAILS – Section 4 through 4.2

Performance: The intent of this standard is to ensure that each user has a reliable experience in the load time of the web page. **Developers should keep in mind that a 10 second load time represents the maximum.** Developers should target response times in the sub second – 2 second range for the majority of their pages. Additionally, by keeping the requested content size within 150,000 bytes, we can provide a reasonable load time even if the user does not have broadband connectivity. Performance is one of the areas that OIS has identified as important for continued customer service and support in our OIS Business Plan.

WebResource.axd, Cascading Style Sheets and .JS Files are excluded because they are cached by the browser and are usually considered a necessary contribution to the page size.

Fiddler may be used to ensure compliance with the Performance standard. It is an approved BSSO download. Fiddler provides download time estimates for various data connections (elapsed time & round trip). Your application should be tailored to the target audience's connection speed.

Horizontal scrolling is undesirable. The screen width of 1024 pixels was chosen because it is the most common screen width used within FDOT. Pages sized to meet this screen width will not have problems meeting this standard at higher screen resolutions/widths. Screen sizing should be considered to match the target audience. For example, if the target audience will be connecting with a mobile device, using relative page sizing may be the optimal design.

Any multimedia embedded within an application is subject to any applicable Multimedia Standards.

5. Printing

- 5.1. Pages must be printable using the browser's print function. (See Section 12.7).
- 5.2. Customized printing ("printer friendly" pages) should not interfere with the browser's print function.

5.2.1.Applications choosing to use an icon to represent a “printer friendly” print option must use the standard printer friendly icon. (See Section 6)

SUPPORTING DETAILS – Section 5

Printing Best Practice: There are three ways to provide printable pages within a web application.

- a) **Design at 600 Pixels.** If your web page is viewable on a screen area of 600 pixels, your users will not have any problems printing the page using the standard default printer, browser and browser settings.
- b) **Use Internet Explorer 7.0.** Internet Explorer 7.0 includes a feature that automatically “shrinks to fit” all printed text. This does not ensure compliance for Internet Applications, since the browsers of external users are outside of the Department’s control.
- c) **Provide Printer Friendly Pages.** Custom Print solutions or “printer friendly” pages provide the user a print feature optimized for printing. This is achieved by removing or reformatting elements on the page such as: navigation, banners, images, headers and footers.
- d) Print preview each of your pages to ensure that they are not truncated on the right margin.
- e) Use the online Width Testing Tool available on the BSSO Web Site. This Tool allows you to enter a URL that you would like to test for compliance with this standard. This tool is available at <http://infonet.dot.state.fl.us/BSSO/WebPageWidthTestTool.htm>

References:

MyFlorida.com Portal Coding & Design Standards available at
<http://dms.myflorida.com/content/download/18272/97768/version/1/file/STO-2-72-006.pdf>

6. Graphics

6.1. The following logos and graphics are located on all application web servers, in the image folder of the root directory. Using “relative addressing” on the web page will allow the image to be accessed from the current server as the application progresses through UNIT Test, SYSTEM Test and into PRODUCTION without having to change any code on the web pages.

- 6.1.1.OIS Logo (filename: OISLogosm.jpg or OISLogosm.png)
- 6.1.2.FDOT Logo (filename: dotlogosm.gif)
- 6.1.3.MyFlorida Logo (filename: myfloridasm.gif, myflorida.png or myflorida.gif)
- 6.1.4.Printer Friendly Icon (filename: print-icon.gif)

SUPPORTING DETAILS – Section 6

The use of common images contained in a centralized location allows the developers to always have access to the most current and correct version of each logo. Additionally, if a logo should change, a centralized location allows the update to occur in one place with little or no impact to the applications.

It is the intent of this standard that the images be used, as provided, without being resized or altered by any means. The png images have a transparent background.

7. Animation

- 7.1. Do not use any blinking or moving fonts.
- 7.2. Do not use animated images.
- 7.3. Do not create the simulation of movement by repositioning images in a web page.
- 7.4. Do not create the simulation of movement created by displaying a series of pictures, or frames.

SUPPORTING DETAILS – Section 7.1

Blinking fonts or moving fonts, animated images, and simulation of movement are not generally seen as necessary

within a Business-related or Enterprise Level application. For this reason, BSSO has chosen to disallow their use. Contributing factors to this decision include:

- a) [Section 508, Subpart B, §1194.22\(j\)](#) requires that “pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz”
- b) The flicker rate is cumulative; therefore multiple moving graphics would increase the Hertz rate of the page.
- c) Animated graphics add unneeded weight to the page and could cause problems with adhering to BSSO’s File Size standards listed in Section 4.1.

8. Copyright and Attribution

- 8.1. Never use text, diagrams, photographs, audio, multimedia, program source code, script or graphics from another author’s web pages unless the author explicitly states it may be freely copied or you make appropriate arrangements with the author.
- 8.2. When copying or paraphrasing information from another source, always make an appropriate attribution.
- 8.3. Placement of credit lines for text or article should be at the end of the source or article
- 8.4. Never hyperlink deep-links to material from another web site or on commercial web sites without giving credit. A deep link is a hyperlink that bypasses a website’s home page and takes the user directly to an internal page.

SUPPORTING DETAILS – Section 8

Copyright is the legal right granted to the owner of the copyright to distribute, make derivative works, or show in public the product of their work. The “work” includes software which is considered to be copyrighted in most countries by default even if it does not contain the copyright symbol identification.

Because the technology of websites allow direct links to a particular web page URL, we must pay attention to the page being linked to in order to ensure that credit is given to information shown by external authors. The credit for the work may have been given on the initial “home page”, but when we directly link, we miss seeing the credit.

The programming work that we do for FDOT is considered the ownership of FDOT.

Best Practice:

- a) Document, by using comments, the source of the program code when obtained from sources not within FDOT.
- b) Deep links can be handled in a couple of ways: either consider linking directly to the page that gives attribution to the author, but then makes the user drill down to what you actually want them to see -OR- use a area of your web page or information boxes to give credit to what you are about to deep-link.

References:

[Dictionary.com](#) basic definition of Copyright

9. Header

- 9.1. A page header is required on each page.
- 9.2. The header must include but is not limited to the following:
 - 9.2.1. Application Identifier
 - 9.2.2. A link to application or page level help, with the exception of the actual Help Pages.
 - 9.2.3. (Internet) The FDOT Logo must be located in the top left corner.
 - 9.2.4. (Intranet) If the FDOT Logo is used, it must be located in the top left corner of the header (see Section 6 Graphics). The FDOT logo is not required.

10. Footer

- 10.1. A page footer is required on each page.
- 10.2. The footer must include but is not limited to the following:
 - 10.2.1. Service Desk contact information must be centered in the footer.
 - 10.2.2. The OIS Logo must be located in the left corner of the footer (see Section Graphics)

- 10.2.3. (Internet) MyFlorida Logo must be located in the bottom right corner (see Section 6 Graphics)
- 10.2.4. (Internet) A link to the Department's privacy policy located at <http://www.dot.state.fl.us/PublicInformationOffice/privacypolicy.shtm> must be included in the application footer
 - 10.2.4.1. The text for this link must be "Internet Privacy Policy, Disclaimers & Credits".

SUPPORTING DETAILS – Section 9 & Section 10

The use of Headers and Footers provides the user with a consistent experience for all OIS developed applications. End Users learn that certain information can always be found in the Header and Footer.

The requirement for Service Desk contact information ensures the user always has a way of finding out how to report a problem with an OIS Application.

Standards relating to the placement of logos (Section 9.2.3, 9.2.4, 10.2.2 and 10.2.3) support the idea that OIS developed applications will be "branded" in a certain way. The logos most applicable (FDOT, OIS, MyFlorida) are allowed, and their location is directed by standard, so that the screen does not become overwhelmed with logos. Unless a location is reserved for a logo by the standard above, application identifiers or office level logos can be used in the space.

It is acceptable for the Application Identifier (Section 9.2.1) to be displayed via text or graphic.

11. Approved Software

- 11.1. The only products approved for OIS web page application development are:

<u>Product</u>	<u>Purpose</u>
11.1.1. Microsoft FrontPage	Classic ASP Code Maintenance
11.1.2. Web Focus	Web Focus Reports
11.1.3. MRE	MRE Reports
11.1.4. Visual Studio .NET	.NET Development Tool
11.1.4.1. MapDotNet	Maintenance of .NET 1.1 GIS Maps
11.1.4.2. Microsoft AJAX Control Toolkit	.NET Development Tool
11.1.5. Hummingbird	Electronic Document Management

SUPPORTING DETAILS – Section 11

Identification of a standard set of languages, tools and technologies for use by BSSO programmers allows staff to maintain multiple applications. The decision to adopt new approved software is made by BSSO and coordinated by others throughout OIS, so that we can ensure all tools fit within the enterprise and are supportable over the long term.

12. Coding Methods and Techniques

- 12.1. Frames must not be used.
- 12.2. Query Strings must be URL encoded.*
- 12.3. URLs must not exceed 255 characters.
- 12.4. Persistent cookies must not be used.*
- 12.5. Server.Script timeout must not be used.*
- 12.6. You must not override the server settings.*
- 12.7. You must not disable the browser features.
- 12.8. Do not use FrontPage generated code.
- 12.9. Absolute URLs must be fully qualified.*
- 12.10. Data Validations must be performed at the server level.*
 - 12.10.1. Client side validations can be used as a supplemental method of validation, but not as the

only method of validation.*

12.11. Javascript must be used for all client side scripting*.

12.12. Confidential or secure data that is passed within or outside of the application must be encrypted.*

*These standards are checked via the Web Application review for ASP applications and in the .NET code review for .NET applications.

SUPPORTING DETAILS – Section 12.1

Frames present a number of difficulties including:

- a) Bookmarks do not work as you would expect; you can bookmark the top-level (frameset) page, but not necessarily what is displayed on your screen.
- b) Frames do not usually print the way the screens look.
- c) It is difficult to restyle content within frames since even simple restyling like increasing text size often results in clipping or the need for horizontal scrolling.
- d) It is difficult for users utilizing voice recognition software to determine what potential changes will occur to all frames when they select a link in one particular frame.

Reference:

University of Illinois at Urbana/Chicago, Campus Information Technologies and Educational Services and Disability Resources and Education Services.

<http://html.cita.uiuc.edu/nav/frame/>

SUPPORTING DETAILS – Section 12.2

Certain special characters have meaning when contained in a query string (spaces, ?, =, &, etc.) and may cause problems for a browser if they are not being used for their intended purpose. To be able to safely pass these characters in a query string, the string must be URL encoded. Both ASP and ASP.Net have built-in function that will encode strings for you.

For ASP use the Server.URLEncode Function

For ASP.Net use the System.Web.HttpUtility.UrlEncode Function

Unencoded string example:

Omar Shaikh <omar.shaikh@dot.state.fl.us>

Encoded string example:

Omar+Shaikh+%3comar.shaikh%40dot.state.fl.us%3e

SUPPORTING DETAILS – Section 12.3

Certain older browsers and handheld devices cannot handle URLs that exceed 255 characters. This standard ensures that all users, regardless of browser type, can access URLs generated by our applications.

SUPPORTING DETAILS – Section 12.4

Persistent cookies may present security and privacy concerns to users. FDOT's stance concerning privacy is covered in our Internet Privacy Policy.

Reference:

Florida Department of Transportation Internet Privacy Policy

<http://www.dot.state.fl.us/PublicInformationOffice/privacypolicy.shtm>

Webopedia Online Computer Technology Encyclopedia <http://www.webopedia.com>

SUPPORTING DETAILS – Section 12.5

The Server.Script timeout setting has been mentioned specifically because of the huge potential for performance degradation that it provides. If scripts are timing out on a regular basis they should probably be re-addressed either by breaking them into smaller tasks or perhaps moving to an offline batch process.

See below for a more general overview of why server settings should not be changed by an application.

SUPPORTING DETAILS – Section 12.6

Applications in OIS generally reside in a shared hosted environment (the unit/system and production servers) and all need to have access to available resources. Most settings on the server are there to ensure that applications do not create an issue for all the other applications that are running alongside it, therefore these settings should not be changed by any single application.

If you have a question on a specific server setting, please email CO-BSSOWebDev.

SUPPORTING DETAILS – Section 12.7

People have an expectation of how a web browser will be setup. Users expect to have a home button, back and next buttons and an address bar. We want to give users the browser functionality they are accustomed to.

This is also an Accessibility courtesy. Users with disabilities often need to change browser settings such as font size and color. Altering the web browser setup takes away their ability to make those changes.

SUPPORTING DETAILS – Section 12.8

Relying on vendor specific items makes code less flexible and can present problems when a new version of the product changes or removes previous implementations.

SUPPORTING DETAILS – Section 12.9

This standard ensures that links are usable by all users, including district, handheld and RAS/VPN users.

SUPPORTING DETAILS – Section 12.10

Validating your data on the server ensures maximum protection from user error and malicious attacks, because the validation logic is not available in the page source and the system is not reliant on the user having client side scripting enabled. This also insulates you from having to code for the differences in certain browser scripting engines.

This standard also facilitates interfacing between different pages and web applications.

It is, however, sometimes useful to perform some of the same validations on the client side to free up server resources by not performing a number of unnecessary calls to the server.

SUPPORTING DETAILS – Section 12.11

Javascript is the most widely used client side scripting language on the web and it has the most support from browsers, handheld devices and assistive technologies.

SUPPORTING DETAILS – Section 12.12

Most data passed between pages is sent using either a GET or POST which puts the data in the query string or in the http request header, neither of which is secure. Sending data in this manner is highly susceptible to being read or tampered with. Encrypting the data before it is sent prevents this since the data is meaningless until it is decrypted and cannot be tampered with.

FDOT Enterprise Library Data Marshaller Component may be used to pass and encrypt data between applications.

13. Naming Convention (this includes directory and file names as they appear in the browser)

- 13.1. Do not use spaces.
- 13.2. Do not use underscores.

SUPPORTING DETAILS – Section 13.1

Some browsers interpret spaces in file names as "%20". Cutting and pasting of URLs with a file name that includes spaces can result in problems for application users.

SUPPORTING DETAILS – Section 13.2

Underscores can easily get lost within a hyperlink. The hyperlink hides that fact that there is an underscore separating two parts of the URL. When people try to retype the URL they can mistakenly put in a space instead of the underscore.

Best Practice:

- If you must visually separate a two-word file or directory name, use a dash (hyphen) rather than an underscore.

14. New or Separate Browser Instances

- 14.1. An ALT or TITLE attribute must be used to indicate the link will open another instance of the browser. The attribute value must include the text "**Opens new browser window**". Additional descriptive text may be included if desired.

SUPPORTING DETAILS – Section 14.1

Although modern screen readers and some web browsers alert users when a link opens a new browser window, old screen readers and some browsers do not. Users with cognitive disabilities may not be able to interpret what happened when a new browser window is spawned.

Reference:

UIAccess.com – Resources for Accessibility <http://www.uiaccess.com/spawned.html#wcag>

Web Content Accessibility Guidelines 1.0 <http://www.w3.org/TR/WCAG/>

15. Security & Authentication

- 15.1. Web Applications requiring authentication must use the login method approved for their development platform. These include:
 - 15.1.1. Hummingbird Applications: Login is authenticated against the Hummingbird DM Server.
 - 15.1.2. Intranet Web Applications:
 - 15.1.2.1. Login is authenticated against RACF using the Standard Security Module invoked either by the Common Login for ASP or the FDOT Enterprise Library Authentication Component for ASP .NET.
 - 15.1.2.2. Login is authenticated against Active Directory using LDAP.
 - 15.1.3. Internet Web Applications:
 - 15.1.3.1. The Security Disclaimer must be displayed as part of the authentication process.
 - 15.1.3.2. Login is authenticated against RACF using the Standard Security Module invoked either by the Common Login for ASP or the FDOT Enterprise Library Authentication Component for ASP .NET.
 - 15.1.3.3. Login is authenticated against the Internet Subscriber Account (ISA) system using the Standard Security Module invoked by the FDOT Enterprise Library Authentication Component for ASP .NET.
 - 15.1.3.3.1. Only non-DOT staff may be authenticated using ISA.

- 15.1.3.3.2. The ISA Terms of Use Agreement signature control that is provided by the FDOT Enterprise Library must be incorporated.
- 15.1.3.4. Applications that require authentication must use Secure Sockets Layer (SSL) and disable HTTP access.
- 15.1.4. Single sign-on
 - 15.1.4.1. Authentication must be established using one of the standard designated methods (as listed above).
 - 15.1.4.2. The FDOT Enterprise Library Data Marshaller Component must be used to pass authentication credentials between web applications.

SUPPORTING DETAILS – Section 15

Standardizing the method of Authentication ensures that users are authenticated against safe, reliable and maintained authentication end points. These authentication sources are also those approved by the Computer Security Administration (CSA) Office. This is especially critical when it comes to applications hosted on the Internet. Lotus Notes and Hummingbird based applications use their built in data stores for authenticating users. For Intranet hosted applications written in ASP or ASP.NET, RACF and Active Directory are the only approved methods. Active Directory authentication is not available for Internet Applications. FDOT's Active Directory is not made available on the Internet due to security concerns.

Applications that require authentication via the Internet by something other than RACF should utilize the Internet Subscriber Account (ISA) System. ISA includes a standard Terms of Use (TOU) document that explains how ISA is used and the subscriber's responsibility. It is required that the subscriber acknowledge that he/she has read and agrees to the TOU.

Computer software and/or hardware can intercept and log traffic passing over the Internet. To help reduce the risk of sensitive information being intercepted and interpreted, Secure Sockets Layer (SSL) is used to encrypt the contents of the HTTP transactions.

The FDOT Enterprise Library Data Marshaller Component provides the required encryption of confidential and sensitive data between applications.

16. Browser Compatibility

- 16.1. Intranet Application: Must be compatible with the version of Internet Explorer currently used in FDOT's standard desktop installation.
- 16.2. Internet Application: Must be compatible with the version of Internet Explorer currently used in FDOT's standard desktop installation, along with the latest release of Internet Explorer. In addition, they must be compatible with the latest release of Mozilla/Firefox, and any other browsers required to support the application's user base.

SUPPORTING DETAILS – Section 16

Mozilla follows the W3C standards and has an increasing user base. Internet Explorer is most often used on the Internet. Because of these reasons, Internet applications must be compatible with both browsers. Internet Explorer is the department's only web browser, so Internet applications written to the version of Internet Explorer currently used by FDOT will work correctly for Intranet users.

17. Change Documentation

- 17.1. All changes made to applications must be documented.
- 17.1.1. Applications that use Subversion source control must complete comments when changes are committed.
- 17.1.2. Applications that do not use Subversion source control must document changes in the ChangeLog.xls file located in the _private folder within the root directory of the project or web site.
- 17.1.2.1. The _private Properties must be set to not allow files to be browsed.
- 17.1.2.2. A standard template is provided, and must be used. This file must include, but is not limited to, the following: PURPOSE, DATE of change, developers name or USERID, File/Component, and Change. See example below. The standard template is available at http://tlbstws.dot.state.fl.us/_private/ChangeLog.xls and http://userappsunit.dot.state.fl.us/_private/ChangeLog.xls.

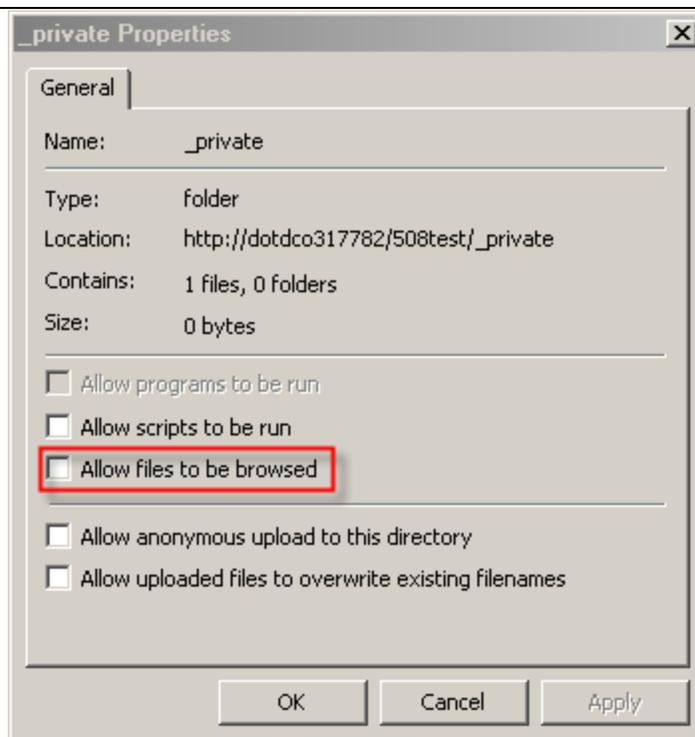
	A	B	C	D	E	
1	Application Name: CITS					
2	Date	UserID	File/Component	Purpose	Change	
3	1/20/2007	SS973HC	Contacts.htm	Updated Contact Information	Added OIS Office	
4	1/20/2007	KN973LN	Default.aspx	Compliance with Standards	Added OIS Logo	
5						

SUPPORTING DETAILS – Section 17.1

The change documentation provides information for the maintenance of the application. By knowing who made the last changes to a Web page or component, we know who to contact if there are questions regarding the changes. Code Documentation includes the practice of placing "inline" comments within the individual code components or regions. The comments describe the purpose and usage of the individual coding regions that are being commented. Code documentation would typically include the purpose of the code, who created the code and any required explanation of how the code should be utilized. This technique assists development staff in better understanding an application, as well as allowing them to more quickly respond to problems or required changes.

SUPPORTING DETAILS – Section 17.1.2.1

To prevent files within the _private folder from being served to the browser the Properties should be set as shown below. The check box "Allow files to be browsed" must not be selected.



SUPPORTING DETAILS – Section 17.1.2.2

The Change Log provided above contains basic information for the maintenance of the application. It describes the who, what, when and where for the last change. The change log can contain additional information, but it must contain the minimum items required in the standard. The **Change Log** is an index or high level listing of changes that have been made to an application. The log would include what parts of the application were changed, who made the change and why the change was made. The change log is maintained in one file located in the root of the application or project.

18. Email Messaging

- 18.1. Email addresses must be fully qualified.
- 18.2. An individual's email address must not be hard coded.
- 18.3. Email addresses must be an existing email address in the Department's email system.
- 18.4. Application-generated emails that may handle responses from the recipient must be produced using a valid from and/or reply to address.
- 18.5. Application-generated emails that do not handle responses from the recipient:
 - 18.5.1. Must be produced using the reply to address of "DoNotReply-FDOTApp@dot.state.fl.us".
 - 18.5.2. Must include a warning to the recipient that the email should not be "replied to" and any responses will not be monitored.
 - 18.5.3. Must have email rules established in the DoNotReply-FDOTApp Inbox to handle server notifications of undeliverable emails.

SUPPORTING DETAILS –

It is important that any email generated from an enterprise application use a valid and active email address in the "From" or "Reply To" fields. If the email system cannot deliver the message (for whatever reason), and the From or Reply To fields are invalid, the delivery failure message has no valid sender address to go back to. This generates unneeded traffic on the email system as the system continues to try and reach the invalid email address. It is also important that emails generated from applications be sent only to valid recipient addresses. When recipient email addresses are no longer valid, the email system continues to attempt to deliver the message. The Functional

Office should have processes in place for updating their application when an address is known to be no longer valid. Steps taken to prevent an application for continuing to send emails to an invalid email address will help reduce the load on the Email System. Establishing an email rule in the DoNotReply-FDOTApps inbox that forwards undeliverable emails will allow the application users to correct the invalid email addresses. Contact the Department's Email System Administrators to setup email rules.

Project Teams should keep in mind that MailTo links will not work if a user does not have an email client installed. MailTo links attempt to open the default email client software and start an email addressed to the listed email. In cases where the browser does not have access to an email client, it may be best to avoid the use of MailTo links, and instead display the fully qualified email address directly on the screen.

STANDARDS CHANGES, EXCEPTIONS AND COMPLIANCE

Requesting an exception or change to the standards.

1. Project Teams may request exceptions or change to the standard.
2. The exception or change requests must be provided, in writing, to the BSSO Quality Assurance Specialist by the OIS Application Coordinator of the Project. The request must include:
 - 2.1. Standard(s) for which they are requesting the exception or change.
 - 2.2. Business case justifying why the exception or change is needed.
 - 2.3. Technical details of the non-standard implementation or the change being proposed.
 - 2.4. Impact to the department for the exception or change.
 - 2.5. List alternatives considered with pros and cons of each alternative.
 - 2.6. Provide justification of why requested exception was the chosen alternative.
3. The request for exception or change will be reviewed by a team assembled by the BSSO Quality Assurance Specialist.
4. The review team will provide a written recommendation to the BSSO Manager.
5. Final decision will be determined by the BSSO Manager.

SUPPORTING DETAILS – Requesting Exceptions or Changes

The Application Development arena is constantly changing. The Application Web Standards must also change to meet the needs of our growing Application Development community. The process for requesting exceptions or changes is the method by which the BSSO Web Standards group is made aware of the possible need for changes (temporarily – as an exception, or permanently – as a standards change). Exception requests should be processed as soon as they are recognized in the Project. Project Teams requesting changes/exceptions are required to provide the listed documentation to assist in the research and understanding of their particular situation.

Compliance Refresh

When a maintenance or enhancement release is scheduled on the BSSO Work Plan, part of the scope of work must include bringing the application into compliance with the latest version of the standards. If the application has any previously granted exceptions, they must now be addressed and made compliant with the documented standard.

When a Web Standards Review is conducted, the application will be reviewed under the identified Web Application Standard for the application as a whole.

SUPPORTING DETAILS – Compliance Refresh

As technology changes, there is a need to continue to update the Web Application Standards. Generally, our Web Application Standards are updated a minimum of twice a year. The intent of this standard is to ensure that our applications are staying current with the recent standards. The Project Team has the option of being reviewed under the standard in place at time of review, or the previous standard.

If an exception was previously granted, and the application cannot be remediated to the new standard, a new exception request must be applied for.