## FDOT Development EnvironmentEffective 01/31/2013

## Technology Requirements

* All source code must be compatible with Visual Studio 2010 Standard or Professional
* All source code will be stored in FDOT’s Subversion repository
	+ Its recommended that the latest code is committed to Subversion at least once a week
	+ FDOT retains the right to request a full code commit at any time
	+ A tagged commit must be done in coordination with all Unit Test publishes
	+ All System Test publishes will be done with builds from tagged versions from the Subversion repository
* All web applications/websites must be set up to run on Windows Server 2003 w/ IIS 6.0
* All code must conform to [Microsoft’s C# Coding Conventions](http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx) and must follow [Microsoft’s .NET Framework Guidelines and Best Practices](http://msdn.microsoft.com/en-us/library/ms184412.aspx) wherever possible
* All code must conform to the current [FDOT coding standards](http://www.dot.state.fl.us/ois/AppDevDocsAndGuidelines.shtm)

## Publishing

* The Vendor will have the authority to publish their code to the project’s Unit Test website
* System Test publishes will be performed by the FDOT Architect from a specific Subversion code version
* Production publishes will be performed by SSRC staff via the FDOT Architect, and will be from the current System Test code
* Batch Job publishes in both Unit Test and System Test will be performed by the FDOT Architect from a specific Subversion code version
* Batch Job publishes to Production will be performed by SSRC staff via the FDOT Architect, and will be from current System Test code
* Publishing requires software that can handle FrontPage Server Extensions (i.e. Visual Studio 2010 or SharePoint Designer)

## Infrastructure

* Unit test – our first level testing environment. This is where developers can perform integration testing to ensure that the application works correctly in FDOT’s server environment and with the other enterprise systems it might be interacting with. All databases typically have an instance at this level to facilitate testing. These databases are typically refreshed as needed from System Test or Production.
* System test – our second level testing environment. This is where the Users will perform their User Acceptance testing and also the staging environment for Production migrations. All databases typically have an instance at this level to facilitate testing. These databases are typically refreshed monthly from Production (for all applications that are live in Production).
* Production – our live environment.
* All web servers are currently Windows Server 2003 w/ IIS 6.0
	+ For MVC projects, extensionless routing is not enabled.
* FDOT uses multiple Active Directory domains that all reside in a single Forest w/ Trust
	+ The FDOT Architect can provide more info related to this if needed

## FDOT Enterprise Library (FEL)

FDOT has a code library that can be utilized by any application for various purposes. Some parts of the library are required to be used by all applications for various functions. Other parts are required to be used if interactions with certain Enterprise systems are required. Additional parts of the library are for some common functionality, but are not required to be used.

The FEL is partially service based. Test implementations of all of the FEL services will be available over the internet to facilitate development when Vendors are coding and not connected to the FDOT internal network.

FEL pieces required to be used:

* Connection strings – all connection strings must be obtained from the FEL connection string provider by connection label. Vendors will need to work with the FDOT DBAT assigned to their project to obtain connection labels to be used by their project. This same approach also applies to user id/password combinations if an application needs an account for some process (i.e. connecting to a 3rd party web service).
* Error handling –custom error handling is applied at the server to all web applications and handles unhandled server side exceptions by logging and sending emails to the configured email groups. Applications must provide the necessary configuration settings for this to work properly.
* Email – any emails being sent from the application must obtain SMTP settings from the FEL library configuration provider so changes across the enterprise do not affect applications individually.

FEL pieces that are required for certain system interactions:

* The FEL is partially comprised of what we call Enterprise Business Objects (EBOs). These are interfaces into various enterprise systems, and are required to be used if your application needs these interactions.
	+ Staff Repository System (SRS) – the repository of all internal and external staff
	+ Transportation Vendor Information (TVI) – database of all FDOT Vendors
	+ DOT Code Tables (DOTCODES) – access to Enterprise Code Tables
	+ Enterprise Document Management System (EDMS) – storage mechanism for all electronic documents
	+ Organization Codes (OrgCodes) – database of FDOT’s organizational units
	+ Financial Project – database of FDOT’s construction projects
	+ Contract Info – database of FDOT’s Contracts
	+ Internet Subscriber Accounts (ISA) – mechanism for non-FDOT users to create and maintain an account to access various FDOT systems
	+ RACF – an internal authentication/authorization system

Other FEL pieces that are optional:

* Helper Functions – there are a handful of functions that are not required but can be useful. One of the most useful is the EnvironmentLevel property. This returns which level the server is that the application is running on and can be mainly used to run code for the test environments that should not run in Production. Other helpful methods include Business Day Calculations (based on FDOT’s calendar), logging, etc. (See documentation for a complete list.)

## Approved Technologies

* Anything not on this list must be approved for use on a per-project basis.
* Any 3rd party libraries must be included via NuGet where possible. If not possible, then the dlls must be included with the source.
* Vendors may not use third-party tools or code which would place a licensing responsibility on BSSO without prior review and approval of the tools or code by BSSO.

**Languages**

* C# 4.0 (32-bit)
* JavaScript
* Project Types (VS Templates)MVC 3.0+ w/ Razor Syntax
	+ This includes everything that is typically included by Microsoft in an MVC project
* ASP.Net Webforms 4.0+
* Console application (.exe) for any Batch Jobs
* Class Library
* Test Project

**Libraries**

* All core .Net Framework Libraries included in the standard .Net installnHibernate if using an ORM
* ADO.Net with parameterized queries if not using an ORM
	+ Stored procedures are not allowed
* Oracle DataAccess Driver (Oracle.DataAccess.dll - version matching the version FDOT is currently using) must be used for all Oracle database access
* IBM DB2Connect Driver (version matching the version FDOT is currently using) must be used for all DB2 database access
* Ninject – dependency injection
* Automapper – object to object mapping
* Glimpse – web site debugger
* Moq – a testing framework for mocking dependencies
* Open XML SDK – Word, Excel, and Powerpoint exports
* iText# - PDF Generation
* NUnit – Unit testing framework alternative to Microsoft’s unit testing libraries
* Log4Net – Logging framework
* Newtonsoft JSON.Net – JSON serialization library
* Sharp Serializer – Efficient multi-format .Net serialization library
* 960 Grid – CSS Framework
* jQuery and any jQuery plug-ins
	+ All non-minified source must be provided
* Any other 3rd party JavaScript libraries are also acceptable, provided that all source is included and license requirements are reviewed and accepted by FDOT (an IRR must be submitted and approved by FDOT through the FDOT Contract Manager)
* Microsoft SQL Server Reporting Services Client for any reports