Department of Veterans Affairs (VA)

Benefits Claims Decision Support System (BCDSS)

Developers Guide



**March 2017**

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2017-03-09 | 1.0 | No Updates | Vasu Rayapati |
| 2017-02-09 | 0.9 | Updated source code repository section [3.1](#_Source_Code_Repository) | Vasu Rayapati |
| 2017-01-20 | 0.8 | No updates |  |
| 2016-12-27 | 0.7 | No updates |  |
| 2016-11-16 | 0.6 | Added instructions and screen shots in Development Tools and reformatted Database Setup | Sukumar Makkapati |
| 2016-10-20 | 0.5 | [Database Setup](#_Database_Setup), [SoapUI](#SoapUI), Commit code changes | Dominic Yeh |
| 2016-09-22 | 0.4 | No updates required | Dominic Yeh |
| 2016-08-23 | 0.3 | Removed NoSQL reference [4.1](#_Implementing_Front-End_Components) | Vasu Rayapati |
| 2016-06-29 | 0.2 | Guide Update | Dominic Yeh |
| 2016-04-20 | 0.1 | Template Created | Erik Rothwell |

Table of Contents

[1. Introduction 4](#_Toc467054186)

[1.1 Purpose 4](#_Toc467054187)

[1.2 Scope 4](#_Toc467054188)

[1.3 Intended Audience 4](#_Toc467054189)

[1.4 Assumptions and Dependencies 4](#_Toc467054190)

[2. Setting up the Local Environment 5](#_Toc467054191)

[2.1 Instructions for Building a Developer Environment 5](#_Toc467054192)

[2.1.1 Development Tools 5](#_Toc467054193)

[2.1.2 Database Setup 17](#_Toc467054194)

[3. Configuring and Using the FTL 19](#_Toc467054195)

[3.1 Source Code Repository Overview 19](#_Toc467054196)

[3.1.1 Tool Source Control Overview 19](#_Toc467054197)

[3.2 Checking Files into Source Code Repository 19](#_Toc467054198)

[3.2.1 Refresh Remote Changes 19](#_Toc467054199)

[3.2.2 Refresh Sandbox 19](#_Toc467054200)

[3.2.3 Check-in and Deliver 19](#_Toc467054201)

[3.2.4 Resolving File Conflicts 19](#_Toc467054202)

[3.2.5 Configuring an External Compare Tool 19](#_Toc467054203)

[4. Development Guidelines 20](#_Toc467054204)

[4.1 Implementing Front-End Components Standards 20](#_Toc467054205)

[4.2 Unsaved Changes Validation 21](#_Toc467054206)

[4.3 Required Fields Validation 21](#_Toc467054207)

[4.3.1 Description 21](#_Toc467054208)

[4.3.2 Initialization 21](#_Toc467054209)

[4.3.3 Applying Field Validation 22](#_Toc467054210)

[4.3.4 Usage Tips 23](#_Toc467054211)

[4.4 Miscellaneous Notes 23](#_Toc467054212)

[4.4.1 Target Screen Resolution 24](#_Toc467054213)

[A. Terminology 25](#_Toc467054214)

[B. References 26](#_Toc467054215)

[B.1. User Documentation 26](#_Toc467054216)

[Attachment A. Approval Signatures 27](#_Toc467054217)

# Introduction

This developer’s guide provides the information required for a new member of the Department of Veterans Affairs (VA) Benefits Claims Decision Support System (BCDSS) product development team to create a local environment with the tools and configuration settings necessary to perform his or her role on the team.

Contractors develop the BCDSS application on their laptops and perform integration testing in the VACI FTL environment. The FTL team must configure a consistent image of BCDSS so that developer testing can occur in a common fashion because the BCDSS team develops the application on local contractor laptops. This Guide includes instructions for configuring a local developer workstation.

Local environments should not have any dependencies on the VA network or the FTL Sandbox to develop, build, deploy, and test BCDSS.

***Note:*** As a living document, this document will undergo continual refinement based on feedback from BCDSS team users and other evaluators.

## Purpose

This manual provides the information required for members of the BCDSS product development team to create a software environment and use the tools necessary to develop, test, manage, and document the BCDSS application. It also provides development guidelines for developers.

## Scope

This document’s scope covers the set of tasks necessary to create a local environment for developing, testing, managing and documenting the BCDSS product.

## Intended Audience

This document’s intended audience includes personnel involved with BCDSS product development, including software development, configuration management, testing, documentation, and other functions that require setting up an environment and using the standard set of BCDSS product development tools.

## Assumptions and Dependencies

To keep the local environment up and running, there exists several dependencies and every tool given below is interdependent on the other. To keep the machine up and running, developers should have this document as a primary reference point if they run into any issues while setting up the environment and read the process mentioned below before starting the local environmental set up.

# Setting up the Local Environment

The selections below detail the instructions for configuring a local environment on a local team-member’s computer. These sections assume that local environments will not depend on the VA network or the FTL Sandbox to develop, build, deploy, and test BCDSS.

## Instructions for Building a Developer Environment

Instructions for how to download and configure all tools required to contribute to the BCDSS application can be found on the BCDSS Confluence site.

### Development Tools

**Java SE Development Kit 8u91**

Java Platform, Standard Edition (Java SE) lets you develop and deploy Java applications on desktops and servers, as well as in today's demanding embedded environments. Java offers the rich user interface, performance, versatility, portability, and security that today's applications require.

Table 1: Java SE Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| **Installation** **Path** | |
| 1 | Create a folder under C: called *dev*. |
| 2 | Create a sub-folder under **dev** called *dev-tools*. |
| 3 | Create a new folder called *Java*. Install your **JDK** and **JRE** into this folder by downloading and running the **java.exe** file. |
| **Set the Local Environmental Variables** | |
| 4 | On the desktop, right-click on the **My Computer** icon and select P**roperties**.  User is redirected to **Control Panel Home**. |
| 5 | On the left side of the screen, select **Advanced System Settings**.    This displays the **System Properties** dialog box.  On the **Advanced** tab, select **Environmental Variables**. This will allow you to create a new variable or edit an existing variable. |
| 6 | After installation, set the environmental variables for **JAVA\_HOME** and **PATH**. The path is mandatory, but the developer must decide where to install. A good directory structure makes the process easier and ensures things move smoothly. The following figures provide examples of how to set variables: |
| **Test Installation Procedure** | |
| 7 | To test whether Java is installed correctly, select the **Start** menu, type *cmd*, and press ENTER. |
| 8 | Type *javac* and *java*, then press ENTER. This will enable you to see if all the options exist. If they don’t exist, you will see an error message, “ ‘javac’ is not recognized as an internal or external command, operable program or batch file”. |

**Ant 1.9.4**

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

Table 2: Ant Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the required version of ant binary zip file from <http://archive.apache.org/dist/ant/binaries/binaries/>. |
| 2 | Extract the downloaded bin.zip file to the specific tool directory. After installation, set the environmental variable for **ANT\_HOME**. The path is mandatory, but the developer must decide where to install. A good directory structure makes the process easier and ensures things move smoothly. Use the same path where the ant.zip file is extracted. |
| 3 | Add the *ant bin path* to the **Path** variable in environmental variables. |
| **Test Installation Procedure** | |
| 4 | To test whether Ant is installed correctly, select the **Start** menu, type *cmd*, and press ENTER. |
| 5 | Type *ant –version* and press ENTER. This will enable you to see the version of ant installed. Otherwise, you will see an error meesage, “ ‘ant’ is not recognized as an internal or external command, operable program or batch file”. |

**Maven 3.2.1**

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting, and documentation from a central piece of information.

Table 3: Maven Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the required version of the ant binary zip file from <https://archive.apache.org/dist/maven/maven-3/>. |
| 2 | Extract the downloaded bin.zip file to the specific tool directory. After installation, set the environmental variable for **MAVEN\_HOME** or **M2\_HOME**. The path is mandatory, but the developer must decide where to install. A good directory structure makes the process easier and ensures things move smoothly. Use the same path where the maven.zip file is extracted. |
| 3 | Add *maven* *bin path* to the **Path** variable. |
| 4 | Create a local repository in local for maven to have all third party libraries at one place and place the oracle jars into the repo. The repo can be created by modifying the local **settings.xml** file in the path **C:\dev\dev-tools\apache-maven-3.2.1-bin\conf\settings.xml**. See next step. |
| 5 | Add the following line to the existing settings.xml to create a local repo: <localRepository>C:\dev\dev-tools\apache-maven-3.2.1-bin\apache-maven-local-repo </localRepository> |
| **Test Installation Procedure** | |
| 6 | To test whether Maven is installed correctly, select the **Start** menu, type *cmd*, and press ENTER. |
| 7 | Type *mvn –version* and press ENTER. This will enable you to see the version of Maven installed or you will see an error meesage, “ ‘mvn’ is not recognized as an internal or external command, operable program or batch file”. |

**Git 2.8.1 64-bit**

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Create a Git account or use an existing account which has the master base code repository branch to be granted access. Developer will receive an email from VHA Innovations with the permissions to the repository for code base access and other project related material. Once developer is granted access will be directed to the project VHAINNOVATIONS/BCDS which will take them to the code base page as shown in Figure 1.

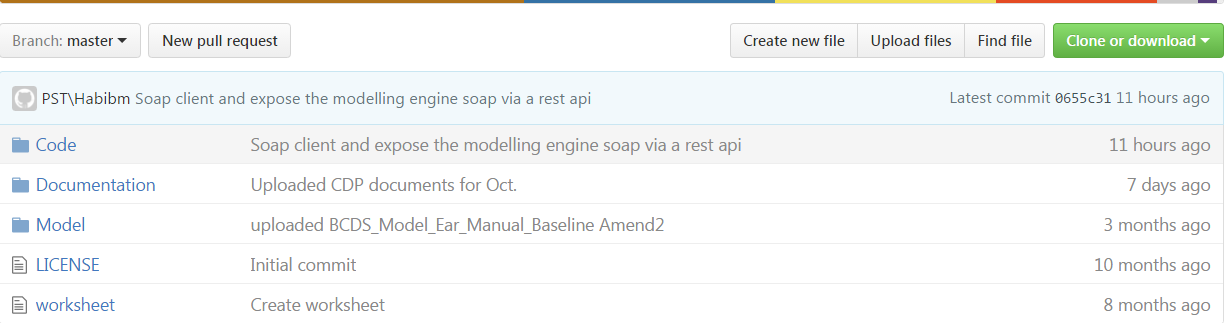


Figure 1: Git Code Base Page

The developer can use the **Clone** or **Download** button to have a local copy of the code base. The developer can use any kind of tool to maintain the versions for accessing Git, e.g., Git Desktop version or another tool like Source Tree. For additional documentation, see:

* <https://help.github.com/desktop/guides/getting-started/installing-github-desktop/>
* <https://help.github.com/enterprise/2.7/admin/guides/installation/>
* <https://github.com/blog/1127-github-for-windows>

**SourceTree 1.8.3**

SourceTree provides you with an interface that gives you the same capabilities you have with Git and Mercurial without needing to type out a bunch of commands.

Table 4: SourceTree Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the **SourceTree Setup file** from <https://www.sourcetreeapp.com/>. |
| 2 | After completing the download, double-click on the downloaded **SourceTree setup file**, which will take you to the installation Wizard. |
| 3 | From the **Atlassian account** page, if you have an account, select **Use an existing account**. A popup appears where you can enter your credentials. If you don't have an account, select **Go to My Atlassian** and follow the prompts on the page that opens to create an account. |
| 4 | Select an option from **Git** **Account** from existing options: **Git** **Account** or **Bit Bucket Account** and link the **Account** to **SourceTree**. |
| 5 | The developer can use the **Pull** operation to get all the delivered changesets into local and see all his local changesets in **Unstaged files**. |
| 6 | To deliver those **Unstaged files**, select **Stage All**. The files in **Unstaged files** will be moved to **Staged files**. The developer can use **Push**, the developer can push his changesets to Git and commit the changesets using **Commit**. The following link has additional required documentation for Git and SourceTree: <https://confluence.atlassian.com/bitbucket/set-up-sourcetree-603488472.html>. |

**Oracle SQL Server 11g (11.2 xe)**

Oracle Database 11g Express Edition (Oracle Database XE) is an entry-level, small-footprint database based on the Oracle Database 11g Release 2 code base. It's free to develop, deploy, and distribute; fast to download; and simple to administer.

Table 5: Oracle SQL Server Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the Oracle Database Express Edition 11g with a specific version compatible to developer local machine from <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>. |
| 2 | Run the **Setup Application file** from **Disk1** and install the database to the existing tools directory through the installation wizard. |

**Oracle SQL Developer 4.1.3 – 64 bit**

Oracle SQL Developer is a free integrated development environment that simplifies the development and management of Oracle Database in both traditional and Cloud deployments. SQL Developer offers complete end-to-end development of your PL/SQL applications, a worksheet for running queries and scripts, a DBA console for managing the database, a reports interface, a complete data modeling solution, and a migration platform for moving your 3rd party databases to Oracle.

Table 6: Oracle SQL Developer Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download **Oracle SQL Developer** from <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>. |
| 2 | Extract the **sqldeveloper.zip** file and follow the installation wizard prompts. |
| 3 | Run the application file (.exe). |
| 4 | Since Java 8 is being installed, a pop-up window will appear. Select **Yes** and continue. |
| 5 | Once the **SQL Developer** is opened, select **Connections** and select **Creating a new connection**. The Username field, type *sys as sysdba*. The password will be the same you provide while installation. Provide all other required parameters to establish a connection as shown below: |
| 6 | Change the configuration in the local configuration file located in **C:\dev\dev-projects\BCDSS\Code\bcds-persistence\src\main\resources\config\liquibase\liquibase.properties** and **C:\dev\dev-projects\BCDSS\Code\bcds-web\src\main\resources\config** where **file application-local.yml** is located. |
| 7 | Modify the existing credentials to the actual credentials provided in SQL Developer while establishing a connection. |

**Eclipse Mars 4.5.2**

Eclipse provides IDEs and platforms nearly every language and architecture. We are famous for our Java IDE, C/C++, JavaScript, and PHP IDEs built on extensible platforms for creating desktop, Web and cloud IDEs. These platforms deliver the most extensive collection of add-on tools available for software developers.

Table 7: Eclipse Mars Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download Eclipse from <https://eclipse.org/downloads/>. |
| 2 | Download the **Java Developer IDE**. |
| 3 | Extract the **zip file** and run the **Eclipse application** (.exe) file. |
| 4 | Create a workspace and name it or use the default location as shown below: |

**Tomcat Apache Server 8.x**

The Apache Tomcat® software is an open-source implementation of the Java Servlet, JavaServer Pages and Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language, and Java WebSocket specifications are developed under the Java Community Process.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project.

Table 8: Tomcat Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the **apache-tomcat-8.5.6-windows-x64 zip file** from <https://tomcat.apache.org/download-80.cgi>. |
| 2 | Unzip the downloaded file and extract it to the tools directory. Server can be ran in multiple ways. Here are few ways to do – |
| 3 | Go to the extracted folder under **Tools** and in the **bin** folder, you should be able to see **tomcat8** and **tomcat8w**. Double-click on the file name and the server will start. |
| 4 | From the **Start** menu, type *cmd* and press ENTER. |
| 5 | Use the **cd** command to navigate to the apache tomcat bin folder as shown below: |
| 6 | Run the **startup.bat** command as shown below:    A new window will pop-up and the server will appear up and running as shown below: |
| 7 | Developers can also configure the server using eclipse. In the top, navigate to **Window 🡪 Preferences 🡪Server 🡪 Runtime Environments**, as shown below: |
| 8 | Select **Add** in **Server Runtime Environments**. A new pop-up will appear as shown below: |
| 9 | In **New Server Runtime Environment** window, under **Apache**, select **Apache Tomcat v8.0**. |
| 10 | Complete the setup by providing the location of the extracted zip file then select **Finish**. |

**Soap UI 5.2.1**

Soap UI is the world most widely used open-source API testing tool for SOAP and REST APIs. Soap UI offers Service-Oriented Architecture (SOA) web services functional testing, REST API functional testing, WSDL coverage, message assertion testing, and test refactoring. With over 10 years of experience backed by a vast open source community, Soap UI is the defacto method for ensuring quality when developing APIs and web services.

Table 9: SOAP UI Installation Procedure

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download the required SOAP UI version from <https://www.soapui.org/Downloads/latest-release.html>. |
| 2 | Go to the **Downloads** folder, run the SOAP UI application file to start the installation wizard. A desktop icon will also exist if required. Utilize this SOAP UI as an IDE for testing the service end-points. |

### Database Setup

Table 10: Add new connections to Dev & Pilot Databases

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | In SQL Developer, use  to add new Database connection. |
| 2 | For BCDS Development Database, set the field values as follows:   * Username: *BCDSS\_DEV* * Password: *developmentonly* |
| 3 | For BCDS Pilot Database, set the field values as follows:   * TST/Pilot Schema: *BCDSS\_TST* * Password: *testonly* |

# Configuring and Using the FTL

## Source Code Repository Overview

The BCDSS source code can be accessed through below URL for authorized users:

<https://github.com/VHAINNOVATIONS/BCDS/>

The content is organized in Code, Documents and Model folder structure.

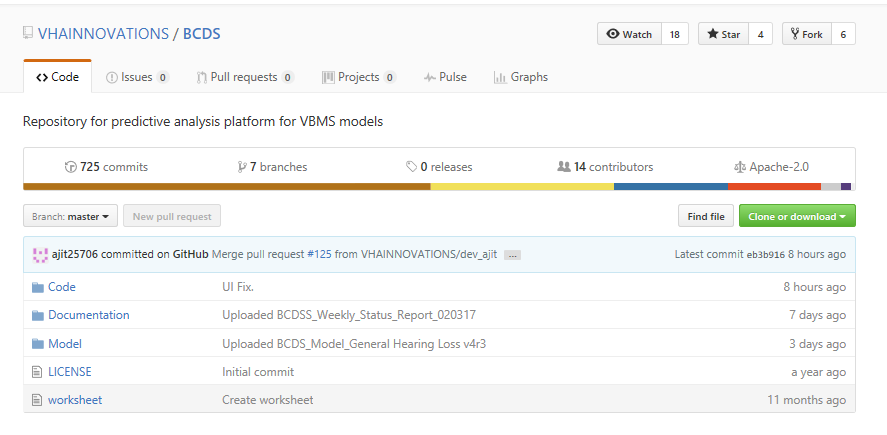


Figure 2: BCDSS Source Code Repository

### Tool Source Control Overview

Users will be able to access Github using SourceTree or GitHub desktop tool for day–to-day operations.

## Checking Files into Source Code Repository

### Refresh Remote Changes

### Refresh Sandbox

### Check-in and Deliver

**Commit Changes to GitHub**

We recommend that committers who have read/write access to the repository do their work on a branch in their local repository:

* Each developer can work on his own complete copy of a GitHub repository stored on his local environment by issuing a “pull” or “clone” command.
* You should always pull the current code base in GitHub down to your local repository, before you can push your new changes to GitHub master branch.
* Once this is done, you can merge your changes with the project’s repository.
* Make sure to add comments to all code commits to the GitHub master branch.

### Resolving File Conflicts

### Configuring an External Compare Tool

# Development Guidelines

These sections provide guidance to developers to ensure a consistent approach for code artifacts.

## Implementing Front-End Components Standards

Adhere to the following standards when authoring front-end components:

* Controllers talk to Services instead of Data Access Objects (DAOs), whereas Services talk to DAOs and Delegates as needed.
* Access the BCDSS Database (BDB) via **Controller>Service>DAO**.
* Scaffold code automatically styles all buttons (<button>) as jQuery buttons by default.
* Place script in <jsp:attribute name=”scripts”> instead of the <jsp:body> area.
* Use <label> for input-element field labels.
* Perform client-side validation for format/type.
* Leverage the showError(…) Java Script (JS) function to populate error messages from validation or server-side errors ().
  + ***Note***: Also use this function for showWarning() and showInfo().
* Some needed styles in BCDSS-app.css will evolve with UX assistance.
* Use Cascading Style Sheets (CSS) to lay out elements (versus HyperText Markup Language (HTML) tables).
* Use the basic rule for Tab Order of: Left-to-Right and Top-to-Bottom. Use the following HTML tabindex ranges to enforce keyboard Tab Order:
  + North Panel—0000-0499 range
  + West Panel—0500-0999 range
  + Center Panel—1000-1999 range (main content for screens)
  + East Panel—2000-2499 range
  + South Panel—2500-2999 range
* Use CSS class ‘priorityFocus’ for input elements that need default focus after page (*or partial page*) refresh. This allows the use of $(‘.priorityFocus’).focus() at any time.
* Ensure that you target Internet Explorer (IE) 11 for testing, which is the VA-supported browser.
* Use the Non-Visual Display Access (NVDA) screen reader to smoke test accessibility concerns.
* openProgressBar and closeProgressBarare handled by global callbacks (*if using jQuery Ajax*).
* Use the $(‘#...’) jQuery notation instead of document.getElementById.
* When checking for error conditions from the server side, return error messages in WS payload; do not check data in with JS or create error messages in with JS, as this keeps JS minimal.

## Unsaved Changes Validation

## Required Fields Validation

The following sections cover the validation of the required fields.

### Description

BCDSS incorporates the jQuery ValidationEngine plugin to inform the user in a visual manner that the user failed to fill in required form fields. By default, validations range from email, phone, and URL, to more complex custom validation using AJAX processing or custom JavaScript functions. In the screen below, the Validation Engine visually warns the user of a failure to fill in the required fields for the page.

### Initialization

The Validation Engine requires the following files to enable:

jquery.js (jQuery)

validationEngine.js & validationEngine-en.js

validationEngine-jquery.css

Find the Validation Engine on the web at <https://github.com/posabsolute/jQuery-Validation-Engine>.

The Validation Engine initializes automatically for every page. The util.js function’s pageSetup function to call the enableFormValidation function to initiate the validation engine (this functionality requires no additional steps):

$("#form.id").validationEngine();

Prevent the Validation Engine from enabling by adding a hidden field in the page for pages with no necessary required field validation, as follows:

<input type="hidden" name=" skiprequiredfieldscheck">

### Applying Field Validation

This section provides the details for applying field validation. Find these code snippets in the file login.jsp. The following list provides examples of field validation:

1. When using it as CSS class type—Recommended for simple validation. It works best with simple text input fields:
   * class=”validate[required]”—Any field with this class requires this input control.
   * class=”validate[required, funcCall[validateUserIdType]]”—Calls required and custom function ‘validateUserIdType()’. Developers can hook custom functions. Refer to login.jsp and login.js.
2. When using it as JavaScript—Recommended for more complex validation where a developer desires manual control over the display of an error prompt. In the case of input fields that dynamically disable or turn invisible, users must close the error prompt linked to that input field manually:

$(‘#input\_id’).validationEngine(‘hidePrompt’); //if you want to remove error message manually.

$('#input\_id').validationEngine('showPrompt', 'This a custom msg', 'load');//3rd argument is for background color: ‘load’ = black, ‘pass’=green and anything else = red.

$(‘#form\_id’).validationEngine(‘hide’);//hide all errors for form.

alert($(‘#form\_id’).validationEngine('validateField',‘#input\_id’) );//validates only one field.

$("#form\_id").validationEngine("updatePromptsPosition"); //update prompt’s position. Useful when re-positioning of prompt is necessary.

1. Pre-defined validators:
   * required: Fails if the element contains no value. This validator can apply to almost any kind of input field.

**Sample usage:**

<input value="" class="validate[required]" type="text" name="email" id="email" />

<input class="validate[required]" type="checkbox" id="agree" name="agree"/>

<select name="sport" id="sport" class="validate[required]" id="sport">

<option value="">Choose a sport</option>

<option value="option1">Tennis</option>

</select>

custom[regex\_name]: Validates the element's value to a predefined list of regular expressions.

**Sample usage:**

<input value="someone@nowhere.com" class="validate[required,custom[email]]" type="text" name="email" id="email"/>

funcCall[method\_name]: Validates a field using a third party function call. If a validation error occurs, the function must return an error message that will automatically show in the error prompt.

**Sample usage:**

function checkHELLO(field, rules, i, options){

if (field.val() != "HELLO") {

// this allows the use of i18 for the error msgs

return ‘Text is not HELLO. Please enter HELLO’;

}

}

<input class=”validate[required, funcCall[checkHELLO]]” type=”text” id=”hello” name=”hello”/>

equals[input\_id]: Check if the current field's value equals the value of the specified field.

### Usage Tips

Use the following tips for the validation engine:

* The page must contain a unique field.id across the page.
* For simplicity and consistency, field.id and field.name should match (except with minCheckbox and maxCheckbox validators.
* Spaces and special characters should be avoided in field.id or field.name.
* Use lower case for input.type; e.g., text, password, textarea, checkbox, radio.
* Validators evaluate from left to right. Use the Ajax validator last e.g.: validate[custom[onlyLetter],length[0,100],ajax[ajaxNameCall]].
* Use only one Ajax validator per field.

## Miscellaneous Notes

This section contains miscellaneous information important to know for maintaining development environments.

### Target Screen Resolution

Mobile devices notwithstanding, plan the target screen resolution for BCDSS as 1024x768 using a 4:3 monitor on a desktop/laptop machine using the approved version of Internet Explorer with Text Size of “Normal.” Developers and testers must ensure there is no horizontal scrolling when using this configuration.

1. Terminology

The following index lists the terminology used in this document.

| **Acronym** | **Definition** |
| --- | --- |
|  |  |

1. References

The following documents supplement the information in this document:

* 1. User Documentation

1. Approval Signatures

This section is used to document the approval of the BCDSS Technical Manual during the Formal Review. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add /es/name by each position cited.

The Business Sponsor and Project Manager are required to sign.

REVIEW DATE:

SCRIBE:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Business Sponsor >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Project Manager >