# Self-Updating Website Architecture

The website will check the SBO site for updates whenever an admin user logs in. If updates are available a banner will appear across the top of the website telling the user an update is available. The user clicks on the update button which triggers a series of events to occur that make updates to the website.

### Website Updater

The WebsiteUpdater class provides the external interface to this subsystem.

#### Updates Availabe

The static method Updates Available returns a boolean indicating whether or not an update is ready to be installed. If an update package exists with a version higher than the version in the website config then it returns true. This is used in the \_updateAvailable.cshtml partial view to display an update panel at the top of the page if an update exists. The update panel displays a message saying an update is available and provides a link the user can click to install the update.

#### Run Updates

When the user clicks on the update website link the controller action creates a new instance of the Website Updater class and calls the Run Updates method. This method loops through all available update packages and runs them within a transaction.

Update package is a wrapper for a folder that contains files and scirpts that can change the system. File updates make a backup of the original file before replacing it with the new file. In the event an error occurs the backups are restored. Database updates use a transaction to enable them to be rolled back in case of an error.

#### Update History

As the update is performed a database record is created that summarizes the update including the version, date run, and a description. A log entry is also recorded for each action performed during the update. In the event of an error the transaction is rolled back and the changes are not recorded.

#### Finalizing the Update

Once all actions have completed successfully the version in the web.config is updated which causes the application to restart. The controller action that initiated the update directs the user to a page that displays the update history from the database.

#### Multiple Updates

The system is designed to handle the possibility of several updates needing to run sequentially in case a user does not sign in for a long time. These updates should run fine.

#### Logging

Log4Net has been setup to record any errors that occur during a transaction.

### Update Package File Structure

Update packages are stored in the Updates folder at the root of the website. Each subfolder of the updates folder is an update package. These subfolders use the update version for their folder name. The Version is an 8 digit number formatted as YYMMDDVV representing Year, Month, Day, and Increment for the day if more than one update is created.

Each update folder contains two folders and a Description.txt file. The description file contains html that describes the nature of the update.

The Files folder contains a folder structure that matches the website root. Any file that needs to be replaced is placed in this folder and will be copied to the corresponding location in the website root. For example, to replace the home page view add a file to the update folder in the direcotry \Files\Views\Home\Index.cshtml. This can be used to replace any file in any folder including bin, content, images, scripts, Views, or even the website root such as favicon.ico.

The Scripts folder contains a list of sql scripts contained in text files. Each script will be executed in order, so be sure to name them so that they will execute in the propper order. Each script is executed within a sql command using the execute non query method.

An example of the file structure for the second update created on December 29th 2014 is shown below:

* Updates
  + 14122902
    - Description.txt
    - Files
      * Bin
        + Dealer.dll
        + Dealer.pdb
      * Views
        + Home

Index.cshtml

Contact.cshtml

* + - * Scripts
        + 01.txt
        + 02.txt