

Roadkill .NET Wiki Engine

Last updated 04/27/2013 - version 1.6.

Contents

- [Installation](#)
- [Hosting on a web garden/web farm](#)
- [Roadkill web.config](#)
- [Users and permissions](#)
- [Creating and editing pages, using tags as categories](#)
- [Adding images and media](#)
- [Multi-language support](#)
- [Markup types and parsers](#)
- [Importing and exporting with Roadkill](#)
- [Searching with the inbuilt Lucene.net search engine](#)
- [Themes](#)
- [Theme style reference](#)
- [Contributing](#)

1. Installation

Pre-requisites

Before downloading and installing Roadkill, make sure you have the following requirements installed/enabled:

- Some basic technical knowledge of Windows
- ASP.NET MVC 3 (from <http://www.asp.net/downloads>)
- IIS 7+ (although it will work with IIS Express and Cassini)

You do not need a database server - you can use SQLite if you choose. SQL Server Compact (SQL Server CE) is also supported.

Roadkill hasn't been tested with Mono, but we'd be keen to find out how to get it working - please use the Google groups forum for any help with this.

Installing

Download the current version which is a ZIP file, and unzip it to a new folder. If you are planning to use Roadkill on your development machine or remote access to another machine, you will need to create an IIS site before installing - **Roadkill does not do this for you in the wizard**.

There is a howto for this on this Microsoft page: <http://support.microsoft.com/kb/323972>.

Once your site is created and the file is unzipped, copy/FTP the unzipped Roadkill files to your website root folder. Now open your site in your browser, for example: <http://localhost/roadkill/>

Roadkill has been designed to work from any virtual path, provided the path is configured as an application.

The installation wizard will then guide you through the installation step by step, and write your web.config settings at the end of the wizard. You can manually tweak your web.config settings if you prefer, guidance on the Roadkill settings can be found on the [web.config](#) page.

If your installation fails at any point, simply change the "installed=true" in the roadkill section of your web.config to "installed=false" to reset the

installation.

Common Issues

Web.config can't be written to

This is probably the commonest problem with the installer. If you're hosting Roadkill on shared hosting, your control panel should be able to change the permissions for the web.config. However the application pool that Roadkill is running as *should* have enough permissions to write to the web.config.

If you are running Roadkill on your home/development machine then the simplest solution to the problem is to run the application pool as LOCALSYSTEM or LOCALSERVICE.

Database exceptions at Finished stage

These are almost always caused by your connection string login details or a missing dependency. The stack trace you will see on the Finished page will give you full information of the problem, if you scroll down. You can also look at the event log if you have access to it.

2. Hosting on a web garden/web farm

Roadkill has a couple of issues that you should be aware of if you are hosting it in a web garden/farm (load balanced multi-server) environment.

Files uploads

As of Version 1.5, Roadkill supports attachment uploads to file paths outside of the web root directory - which can include UNC paths if needed. All files are served via a `HttpHandler`.

Caching

Roadkill uses basic object caching for its caching strategy, along side browser caching. This saves trips to the database and is important for text content for mid-high traffic websites. By default Roadkill uses `System.Runtime.Caching`'s default cache, which is similar to ASP.NET's cache. This *doesn't* scale across web servers, so should be turned off if you are using multiple servers.

If you wish to scale to more than one server please contact us via Bitbucket, and a new version can be released fairly quickly to support scaleable cache such as Azure or Couchbase. The object cache is based off the pluggable `System.Runtime.Caching` architecture making this a simple change to make.

3. Roadkill web.config

This page contains reference information about the web.config settings for Roadkill, and expects you to have some knowledge of .NET XML configuration files/web.config files.

Roadkill section

Roadkill stores almost all of its configuration settings in the web.config file. This is written to during the installation, based on the settings you provide.

The web.config file contains a custom roadkill section which contains the bulk of the settings, below is an example:

```
<roadkill
  adminRoleName="Admin"
  attachmentsRoutePath="Attachments"
  attachmentsFolder="~/Attachments"
  connectionStringName="Roadkill"
  dataStoreType="SqlServer2005"
  editorRoleName="Editor"
  installed="true"
  isPublicSite="true"
  userManagerType=""
  ldapConnectionString=""
  ldapUsername=""
  ldapPassword=""
  logging="none"
  logErrorsOnly="false"
  ignoreSearchIndexErrors="true"
  resizeImages="true"
  userManagerType=""
  useHtmlWhiteList="true"
  useWindowsAuthentication="false"
  useBrowserCache="true"
```

```
useObjectCache="true"  
version="1.6.0.0"  
</>
```

adminRoleName the role name for editors. This can also contain one, or a comma separated list of Active Directory groups that are editors if Windows authentication is enabled.

attachmentsRoutePath

This is the virtual path used to serve attachments, which by default is "Attachments". Do not include any trailing or beginning slash, for example for the url /myfiles/uploads/" use "myfiles/uploads".

attachmentsFolder This is the directory that images and files are uploaded to. This can be an absolute file path (including a UNC path), or if the value starts with "~/ then the website root is used.

connectionStringName the name of the connection string from the connectionStrings section, defaults to "Roadkill".

dataStoreType

The kind of database to use. This defaults to SQLServer2008, valid values are:

- MySQL
- Postgres
- MongoDB
- Sqlite
- SqlServer2005
- SqlServer2008
- SqlServer2012
- SqlServerCe

These are not case sensitive. If you are using a custom repository (one you've written yourself), this should equal the full type name of the repository. This feature will be supported in future versions.

editorRoleName the role name for editors. This can also contain one, or a comma separated list of Active Directory groups that are editors if Windows authentication is enabled.

installed true/false. If false, the installer wizard will display when you visit the site.

IdapConnectionString If windows authentication is enabled, this should be the connection string to the Active Directory LDAP server, including LDAP:// at the start of the string.

IdapUsername If windows authentication is enabled, this is the username used to connect to the Active Directory server, which is typically a service account.

IdapPassword If windows authentication is enabled, this is the password for the username used to connect to the Active Directory server.

ignoreSearchIndexErrors *(optional)* true/false. Whether search index errors are ignored when pages are created or edited. Defaults to true.

isPublicSite *(optional)* true/false. If set to false, then all pages require a log in to view.

resizeImages

(optional) true/false. This setting will cause images that are too big to be automatically resized, using a jQuery plugin. Defaults to true.

userManagerType *(optional)* if set, this class (which should inherit from Roadkill.Core.UserManager) will be used for authenticating users. The format of this setting should be "Namespace.Type, Assemblyname" or a similar valid type string. See the Users and Permissions page for details on creating your own implementation.

useHtmlWhiteList *(optional)* if set, this will strip all HTML tags/attributes from the markup of each page (not the Theme HTML, just the markup converted from the Wiki markdown). Tags which exist inside App_Data/htmlwhitelist.xml file will be ignored. True by default.

useWindowsAuthentication Whether Windows Active Directory authentication should be used.

useBrowserCache Enables caching on the browser. This will tell the browser to cache Roadkill pages, and check for updates (If-Modified-Since) each time the page is loaded. If no update is found, no content is returned from the server. Recommended to increase the responsiveness of the site.

useObjectCache

Enables Object caching. Roadkill caches all pages and content in memory after retrieving it from the database. The cache is automatically updated when a page changes. Recommended to increase the responsiveness of the site. Object caching is not currently supported for web gardens or web farms.

version

The current version installed. If this version is less than the current build version, then an upgrade is assumed and the upgrade screen appears. This value takes the format major.minor.0.0 (build numbers are not included in Roadkill), for example "1.6.0.0".

The connection string section will look this for a default roadkill installation:

```
<connectionStrings>
  <add name="Roadkill" connectionString="(your connection string)" />
</connectionStrings>
```

There is usually no reason to change the name of the connectionString in this section as it ties in with the connectionName setting inside the roadkill section.

Settings that are stored in the database

The following settings are not stored in the roadkill web.config section but in the database, and are not critical for the running of the application but preferences.

- AllowedFileTypes
- AllowUserSignup
- EnableRecaptcha
- MarkupType
- RecaptchaPrivateKey
- RecaptchaPublicKey
- SiteUrl
- Title

- Theme

Changing the language of the site

If you want to force the language of your site to something other than the one installed on the server, you can do this using the tag inside the `<system.web>` section. This element is included in the Roadkill web.config, the example below shows how to force the site to use Spanish (Spain), a full list of valid locale names can be found at [\[url:http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo\(v=vs.71\).aspx\]](http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo(v=vs.71).aspx)

```
<globalization uiCulture="es-ES" culture="es-ES" />
```

Email server settings

If you have setup the Roadkill installation to allow signups from users, you will want to setup a mail server that the signup and lost emails are sent via. This can be done via the system.net section which is included in the Roadkill web.config by default, but is configured so that all emails are written as files to a drop folder. Below are the default settings, full details can be found on MSDN [\[url:http://msdn.microsoft.com/en-us/library/w355a94k.aspx\]](http://msdn.microsoft.com/en-us/library/w355a94k.aspx)

```
<!-- Change these settings for signup and lost password emails -->
<system.net>
  <mailSettings>
    <smtp deliveryMethod="SpecifiedPickupDirectory" from="signup@roadkillwiki.org">
      <specifiedPickupDirectory pickupDirectoryLocation="C:\inetpub\temp\smtp" />
    </smtp>
  </mailSettings>
</system.net>
```

4. Users and permissions

Permissions and roles

Basics

Roadkill has just two types of user roles: editors (or standard users) and admins. The only difference between the two is that admin users who belong to the admin role can access the "site settings" page, delete and lock pages.

Authentication

By default, Roadkill uses the database to store its users and passwords. For various reasons it does not use the ASP.NET AuthenticationProviders, but its own security system. The system uses the username field for the page history, however the login always uses the email address, and the minimum password length is 6 characters. You can enable or disable user signups as an admin under the "site settings" page. If you disable user signups, then all users will need to be created via the "site settings" page, where you can also set their password. There is no notion of an anonymous user in Roadkill, all edits and new pages need a user login to do so.

If you enable user signups then anybody can create a user (editor) account on the wiki and create and edit pages. You can restrict pages to so they are only editable by admins however, more details on this can be found on the [Creating and editing pages, using tags as categories](#) pages.

If your wiki is available to the public via the internet, it's recommended that you enable the Recaptcha option. Recaptcha (<http://www.google.com/recaptcha>) will go some way to stop spam bots from creating new logins on your wiki. To enable this, you will need to create an account on the Google Recaptcha page, and copy the API key from there into the Roadkill wizard or site settings page.

Roadkill *does* require new user accounts (if user signups are enabled) to confirm their account via an email address as an extra security measure, so you will need to configure an email server that emails are sent via, in your web.config (see the [Web.config](#) page for details), if you decide to allow user signups.

Active Directory authentication

If you want to run Roadkill inside a Windows based network that uses Active Directory for user logins, Roadkill supports this out-of-the-box. The install wizard will guide you on how configure the various options available and does include fairly detailed instructions.

It's worth downloading AD Explorer ([url:http://technet.microsoft.com/en-us/sysinternals/bb963907.aspx](http://technet.microsoft.com/en-us/sysinternals/bb963907.aspx)) first before configuring the Active Directory options in Roadkill, as it makes it easier to discover the groups and LDAP:// settings required. You are able to set multiple AD groups for the editor and admin roles by separating each group with a comma.

You will need a login that can read from the Active Directory in order for Roadkill to authenticate correctly. This login should ideally be one that does not have a password expiry, but at the same time is not able to login as a desktop user, sometimes known as a service user. Roadkill does not use the application pool login for AD authentication via pass-through auth.

Writing your own user manager

If you want to use an external data source for authenticating your users, this is fairly straight forward to do by implementing the Roadkill.Core.UserManager class. This is an abstract class found in the \Domain\Managers\Security folder in the source tree. The default SqlUserManager is a good place to start for seeing how the database user authentication is implemented.

Once you've written your authentication class, you will need to configure it via the Roadkill [Web.config] settings. The setting is userManagerType, which takes the format userManagerType="Yournamespace.Classname, Yourassemblyname"

Adding new users

If you're using the default database user authentication, you can add new users via the admin "site settings" page in Roadkill, which needs an admin login. When you add a user through this interface no email will be sent to the user, and they will be added to the system without needing to confirm their login.

If you are using Active Directory for authentication then you cannot add new users to the system.

Editing users

As with adding new users, editing users is also done via the "site settings" page. You can use this interface to change people's passwords if needed,

which can also be used to de-activate users if needed.

If you are using Active Directory for authentication then you cannot edit users to the system.

5. Creating and editing pages, using tags as categories

Creating a new page

To create a page in Roadkill, you need to have an editor or administrator login - you cannot edit or create pages anonymously. There are various ways of setting up user rights on your site, which can be found on [Users and permissions page](#).

To create a page, click on the "New page" link on the navigation bar (left side for the media wiki theme). You will then need fill the title for the page and some text content. The format of the text is dictated by the markup type for the site, by default this will be Creole wiki markup which is a more consistent form of the media wiki markup you find on sites such as wikipedia.

The editor is not WYSIWYG, but will insert the tokens/markup for bold, italic, underline, headings, lists, links and images. The question mark button gives you help for the accepted special tokens for the current markup type.

If you are an administrator, you have the option of locking down the page via the "This page can only be edited by administrators." checkbox. This will ensure editor logins cannot edit the page, or revert it to previous versions. This is useful for important pages on your site, such the homepage.

You can embed images into your page using the image button, which launches a file explorer in a new dialog allowing you upload images. By default, Roadkill strips all dangerous HTML from the text which includes script tags, style tags, iframe tags.

You can preview your changes before saving, which displays your page inside a new dialog using the same display engine as the Roadkill site. Once you're happy with this, you can save the page.

Roadkill allows you to categorize your pages using the tags textbox. You can enter relevant tags (which will auto complete if the tag already exists) for the page - more on this below.

Editing a page

Editor logins and administrators can edit any page in the system, changing the title, tags and text content. This works the same way (and uses the same interface) as creating a new page.

Page version history and reverting

Every page edit is stored as a new version in Roadkill, which allows you to revert back to previous versions of the page. This can be done via the "View history" link that is available on every page.

The history page shows every version of the page, clicking on a particular version will show the differences between that version and a previous version. Green highlighted text indicates new additions to the page, red highlighted text shows deletions to the page. From the main history page you are able to move back to a previous version via the "revert" link, this will however create a new version of the page itself.

Tags (aka categories)

Roadkill differs slightly from other wiki engines in that it approaches categorizing pages from a blog engine perspective, by using tags rather than categories. Obviously the difference can be seen as just semantics, but tagging allows you use as many tags as you like for each page, giving a page multiple categories. These tags are also searchable in the Roadkill search interface, more on this on [Searching with the inbuilt Lucene.net search engine] page.

There is one special tag built into Roadkill which is the "homepage" tag. This can be used on one page only (the first page is always used), and tells Roadkill that this page should be used for the homepage on the site.

6. Adding images and media

Roadkill allows you to upload images and other files which you can reference in your pages, or in the case of images, show on the page.

To upload files, click on the image button when adding or editing a page. This will display the file manager in a new dialogue which lets you upload new files and create folders. Deleting files is not currently supported in the file manager.

You can reference existing images by clicking on the image button, finding the image you want to show on the page and clicking the bottom right "select" button. This will display a small preview of the image first.

When you add an image, the markup will automatically be added to the content textbox for your page. You will probably need to tweak this for your needs, for details on the markup, see the help for each markup type via the question mark button

Linking to uploaded files

You can link to uploaded files by using "attachment:" followed by the full path to the file inside the link tag. This should include a slash at the start. For example in the Creole markup:

```
{" This is my Excel file: [[attachment:/Spreadsheets/Sheet1.xls|excel file]] "}
```

Attachments folder

Files are uploaded to the folder you set in the installer, which you can also change as an admin under Site settings->configuration. This defaults to ~/Attachments - you should make sure this folder is not inside the App_Data or bin folders, as the images/uploads will not display in the browser as content cannot be served from these folders.

The upload folder will need read and write access by the application pool that Roadkill is running under. Usually you don't need to worry about this as the process will have write access by default.

The files you can upload are limited by their extensions, which is configurable through the admin site (Site settings->configuration). You may find that

some file extensions will still return "403.1" errors - this is caused by the file extension having no mimetype mapping in IIS. If you map the file type then the file will then display correctly in the browser.

Web farms/garden support

The uploads in Roadkill are not currently synced across servers. Version 1.5 of Roadkill will allow you to use a shared folder for uploads, but for now you will need to manually copy and sync files between the multiple upload folders.

7. Multi-language support

Roadkill supports multiple languages, also known as localization, but currently only contains an English (British/American) word and phrase set for the installer and site-wide labels/help. The validation *bold* is multi-lingual however.

Any help with translations to other languages would be a great help, it involves translating around 400 very basic words or short phrases in a text file - please contact us via Google groups (details on the main page) or issue tracker if you help.

By default the language is automatically taken from the server or machine you have installed Roadkill on. You can change this web.config by changing the culture:

```
<globalization uiCulture="auto" culture="auto" />
```

The uiCulture attribute should be a value from the "culture name" column from this page [url:http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo(v=vs.71).aspx)

Multi-language support for the pages you create will still remain a manual process. The engine has no way of switching languages, but it is possible to use tags to distinguish between page languages.

8. Markup types and parsers

Roadkill enables you to edit your Wiki in three different flavours of markup language:

- [Creole Wiki \(used on with Codeplex.com\)\)](#)
- [Markdown \(used on Stackoverflow.com\)](#)
- [Media Wiki format \(used on Wikipedia\)](#)

It is recommended that you keep with Creole Wiki format as this is best supported by the platform. Only a subset of the Media Wiki features are supported, however the Markdown support is fairly comprehensive as it uses the same parser as the one used by Stackoverflow.

Unsafe HTML removal

Your text will have "unsafe" HTML removed from it when it's displayed (it's still saved however). This is done to protect your wiki site from XSS attacks such as embedding Javascript into the page. Roadkill removes the following HTML tags:

- script
- style
- iframe
- frameset
- applet
- any src/href with javascript
- all Javascript attributes for tags, i.e. onclick, onblur

Object tags *are* supported, allowing people to embed flash movies or other plugins if needed.

Tags that aren't supported

There are several media wiki tags that aren't supported, notably macros, namespaces and advanced table support.

Extra tags

The {TOC} tag is supported by Roadkill, which allows you to insert a table of contents anywhere in your page. The table of contents is generated from all the headings on the page, and automatically adds anchors to the page.

You can define your own custom tags to Roadkill by editing the AppData/Tokens.xml file. This XML file lets you define a tag to search for, and the HTML to replace it with. The file uses regular expressions to search and replace the page content for the tags you define.

Roadkill comes with two tags by default:

```
@@alertbox:your text@@  
@@warningbox:your text@@
```

These display your text in a red and yellow box respectively. The XML also contains a section to enter some help text for the tag. Every token is displayed in the help dialog when you click on the "?" help button on the edit/create page along with this help text.

9. Importing and exporting with Roadkill

Importing

Roadkill currently has one option for importing existing content, which is from a Screwtorn 2 installation. Future versions of Roadkill will include a REST API for importing data, but for now this is the only import option.

To import from a Screwtorn database, you need to login as an administrator and head to the site settings->tools page. From here, enter your Screwtorn database connection string (non-database installations aren't supported) and the tool will import the content including page history and categories into your Roadkill instance.

Exporting

Roadkill supports 3 types of exporting:

- Exporting pages and content as an XML file
- Exporting all image/media uploads
- Exporting each page as text (.wiki) file.

You need to login as an administrator and head to the site settings->tools page to export. All three export types will supply you with the content in .zip format, the .wiki text file export will include the page tags at the top of the file and then the content itself.

10. Searching with the inbuilt Lucene.net search engine

Introduction

The Roadkill search engine is powered by Lucene.net : [\[url:http://incubator.apache.org/lucene.net/\]](http://incubator.apache.org/lucene.net/) . The index is updated each time a page is created or edited. Roadkill indexes the full content of the page along with the following fields:

- id (the unique number each page is assigned)
- content
- title
- tags
- createdby (the username of the person who created the page)
- createdon (the date the page was created)

The search engine displays the first 150 characters of each page's content to display in the search summary. When you search in the textbox, the content field is searched by default. You can restrict the page to certain fields using the usual syntax found on search engines like as Google. For example:

- title:"my page"
- tags:dotnet, vb createdby:editor

Index storage

Lucene stores index files for its search data in the App_Data/Search folder. The worker process/application pool will have rights to read and write from this folder by default, so there shouldn't be any issues with permissions providing the website root folder permissions are setup correctly.

The folder will contains 10 or more files in it, which Lucene may or may not contain file locks for while the site is running.

Common issues

Issues when a page is updated or created

Occasionally the Lucene index gets out of sync with the site from an error or other reason. This can also happen when a page is deleted - the whole index is updated when a page is created or deleted. If this happens you should login as the administrator for the site and head to Site settings->Tools and use the "Rebuild search index" option.

This is the best thing to do for any search related problems, as it 99% of the time it will resolve the issue you're having.

11. Themes

Roadkill supports a fairly comprehensive theming framework that allows you to completely reskin your wiki if needed. The Roadkill installation comes with two themes:

Media wiki

This theme emulates the website style you see with media wiki installations. It contains a navigation bar on the left side of the screen and the logo in the top left.

Blackbar

This is the default theme with a black navigation bar across the top of the page. This theme and some roadkill UI elements in general has been designed for a page with a white background. The theme has a very small ego and has no Roadkill logo on the screen.

Creating a theme in 3 steps

1. Create the theme folder

The first step for theme creation is to create a new folder under the "Themes" folder. Name this folder with the name of your theme, without spaces.

2. Copy an existing theme across

It's easiest to get started by simply copying the "Theme.cshtml" and "Theme.css" files from the "Blackbar" theme folder to your new theme folder. This way you have a skeleton to work with, and can just strip out the parts you don't want.

3. Edit the theme.css

Roadkill has a set of global CSS classes and IDs that you can use in your theme, which are listed below. Theme specific styles are declared inside the theme's "Theme.css" file. This includes font styles, heading styles, anchor styles and tag cloud styles.

12. Theme style reference

When creating a theme, you get the following CSS classes available to you so the site adheres to the DRY principle:

- **.biggest** An font size for long sighted.
- **.bigger** Even bigger font size (1.2em)
- **.big** Slightly (1.1 em) larger font
- **.normal** The default browser font size
- **.small** Slightly smaller than the default font size
- **.smaller** Even smaller font size
- **.smallest** A wee little font
- **.bold** Bolded font weight
- **.italic** Italicised font
- **.dimmed** Sets the opacity of the element to 50% (example)
- **.left** Floats the block level element to the left.
- **.right** Floats the block level element to the right.
- **.rounded** Gives the element rounded edges, in Chrome/Safari and Firefox.example
- **.rounded5** Gives the element 5px rounded edges, in Chrome/Safari and Firefox.example
- **.rounded10** Gives the element 10px rounded edges, in Chrome/Safari and Firefox.example
- **.blueborder** Gives the element a blue border
- **.lightborder** Gives the element a light grey border like the borders found on the edit and history buttons.
- **.clear** Clears both left and right floating elements (clear:both)
- **.wikitable** All user tables are assigned the .wikitable class.

There are also a number of IDs declared globally, so when creating a theme you may need to check the roadkill.css file to ensure there are no name clashes. If in doubt, it is best to use "#yourtheme-idname".

Roadkill uses Twitter Bootstrap to define the following styles globally, rather than on a per-theme basis:

- All tables, for example the table on the page history page
- The edit, history and page information buttons

- Modal dialogs
- The edit page textboxes and buttons
- The login page textboxes and buttons
- Print style sheet

Your theme should include a print style sheet. You don't normally need to create this, copying the Theme.Print.css file from the Mediawiki theme will be enough as the style sheet simply turns off all navigation and headers.

13. Contributing

There's various ways you can contribute to Roadkill:

Improve the documentation

You'll need to be familiar with [Markdown](#) to do this, the documentation is held as a set of .md files in source control, under /docs/source. MarkdownPad is a great free tool for editing the files.

Create a new theme

See the previous Themes sections on how to do this.

Find bugs

..and report them via Bitbucket. But bare in mind the project has some fairly comprehensive automated test coverage (over 500 tests), so a bug hunt is only likely to discover a few unknown issues.

Suggest new features

Please do do this via the Google groups forum.

Write some code

Contribute a patch or have a chat on the Google groups about adding a new feature.

The project doesn't use Codeplex team memberships for commits - we use the brave new world of DCVS where you can clone the repository, make some changes, and then push your changes to our repository (a 'pull request'). We can then diff the changes, and accept or deny them.

If you want to write some code, please stick to the following guidelines for your contributions:

- Use the existing style of code found in the source base - new lines for curlyies, '_' prefix for private members, plus the standard .NET framework

guidelines on naming.

- Tabs for indentation (4 spaces per tab).
- If you do have a different coding style to this don't worry, the code is likely to be auto-reformatted anyway (using Visual Studio's built in shortcut key). Don't take this personally if it happens - it's about consistency rather than my-style-is-better-than-yours.
- Use XML documentation for public classes, properties and methods, and where needed private methods too.
- **No vars!** I get religious about var abuse, in my view it makes code harder to read and unless used for [its intended purpose](#) - where the left side is redundant, dynamic LINQ or complex generic declarations (Dictionary of T for example), then please don't pepper the code with them.
- Add unit tests for the code where possible, which can be a unit test, integration test or acceptance (Selenium-based) test.
- Use constructor injection for dependencies, and add these into the DependencyContainer class (this makes testing and mocking a lot easier).
- Uncle Bob's approach to coding and KISS is the most important factor for the project, it should be easy to read instantly.

