St Andrews Website

Contents

[Document Control 1](#_Toc341131297)

[About this document 1](#_Toc341131298)

[Website architecture 2](#_Toc341131299)

[Deployment Model 2](#_Toc341131300)

[Development Model 3](#_Toc341131301)

[Security Model 4](#_Toc341131302)

[Administration 5](#_Toc341131303)

[Google App Engine 5](#_Toc341131304)

[Google Analytics 9](#_Toc341131305)

[Google Webmaster Tools 14](#_Toc341131306)

[Source structure 20](#_Toc341131307)

[Folder Contents 21](#_Toc341131308)

[Development 24](#_Toc341131309)

[System requirements 24](#_Toc341131310)

[Developer Workstation set-up 24](#_Toc341131311)

[Updates and publication 26](#_Toc341131312)

# Document Control

|  |  |
| --- | --- |
| Title | St Andrews Website |
| Status | Draft |
| Last Modified |  |
| Version |  |
| Author(s) | Antony Cartwright |

# About this document

This document contains a description of the current St Andrews website, the tools available to manage and monitor the site and the tools and processes for updating the site.

# Website architecture

The site is a Java Web Application which can be run either on a single workstation or deployed to a public server. With the exception of the domain name provider, the entire infrastructure used to host and manage the site is provided by Google free of charge.

## Deployment Model

The Deployment Model shows the components that are used to present and manage the running web site application.

Analysis tools

Deployed application stack

Go Daddy – standrewshalifax.org.uk

Google Analytics

Google Apps (Mail and Sites)

Google Webmaster Tools

St Andrews Halifax Web App

Google App Engine

### Go Daddy

Go Daddy is the organisation that resolves standrewshalifax.org.uk to a specific address depending on usage. Both email and web traffic are routed to Google Apps

### Google Apps

Google Apps is a suite of productivity applications which we use to host both the @standrewshalifax.org.uk email accounts (Google Mail) and link the registered web site domain [www.standrewshalifax.ork.uk](http://www.standrewshalifax.ork.uk) to the site on Google App Engine (Google Sites).

### St Andrews Halifax Web App

The website code including the site structure built from Struts Tiles and Java Server Pages (JSPs), the content in HTML. Source code is compiled on a local workstation before either running in a local machine or being deployed to a hosting provider.

### Google App Engine

Google App Engine is an application hosting service which supports our application and many other variants of a web application.

### Google Analytics

Google Analytics provide traffic tracking tools which show how the website is being used, from where and how often.

### Google Webmaster Tools

Google Web Master Tools are a collection of site quality and optimisation tools.

## Development Model

The Development Model shows the components involved in updating and publishing the site.

Development workstation

Remote services

St Andrews Halifax Web Site Source

Java

Maven

Google App Engine

Google Code

Subversion

Google App Engine (SDK)

### Java

Java is the core technology used to build the site. There only 5 Java files in the source code so Java skills are not likely to be required for day-to-day updates.

### Maven

Maven is a build automation tool which runs from the command line (e.g. “Command Prompt” on Windows or “Terminal” on a Mac or Linux)

### Struts

Struts is a web framework which provides libraries to structure the site and create re-usable components. The most noticeable place this is used is for Struts Tiles which are used to maintain the site top level page structure by including a standard header, footer and left navigation.

### Subversion (“SVN”)

Subversion is a version control system designed for source code. Changes are committed to a central repository so many people can collaborate on a single application. Each file has it’s full history of changes tracked.

### Google Code

Google Code is a free version control provider who includes support for subversion. As a condition of providing a free service source code managed in Google Code must be available for other’s to share.

## Security Model

Protection against malicious modifications to the site is through username / password authentication. Accounts are currently shared so no user level auditing is possible. The web application source code is publicly available so it is important that no sensitive information is checked into the source code repository. Passwords must be maintained separately.

There are several accounts which are applicable to one or more service. The structure below shows which services are authenticated using which accounts:

* Go Daddy account (standrewshalifax)
  + Go Daddy - <http://www.godaddy.com/>
* Google account ([standrewshalifax@gmail.com](mailto:standrewshalifax@gmail.com))
  + Google Mail - <https://mail.google.com/>
  + Google App Engine - <http://appengine.google.com/>
  + Google Analytics - <http://www.google.co.uk/analytics/>
  + Google Webmaster tools - <https://www.google.com/webmasters/tools/>
  + Google Code - <https://standrewshalifax.googlecode.com/> (separate SVN password)
* Google Apps administration account (standrewshalifax)
  + Google Apps - <https://www.google.com/a/standrewshalifax.org.uk/>
* Google Mail contact email account ([contact@standrewshalifax.org.uk](mailto:contact@standrewshalifax.org.uk))
  + Google Mail - <http://webmail.standrewshalifax.org.uk/>

# Administration

Website administration in this context is limited to switching between website versions, viewing analytics data and viewing website site performance.

Google Apps includes a plethora of features not covered in this document, the most useful of which is the ability to create new accounts so for email addresses. To investigate Google Apps start with an overview like this: <http://www.youtube.com/watch?v=kJT3pagjd8s>

The Go Daddy domain name service can be used to re-point some or all of the services associated with the standrewshalifax.org.uk domain to alternate providers. This task is not covered in this document. There is help on the Go Daddy site but generally caution is advised as domain administration assumes a reasonable amount of background knowledge.

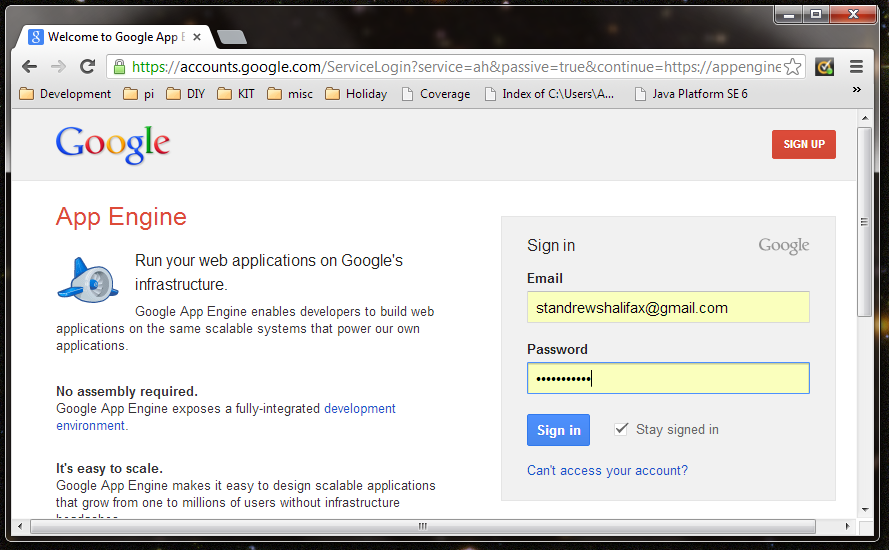
## Google App Engine

Google app engine is used to host the St Andrews Halifax Java Web Application. With the same account it is possible to host other web applications built using other technologies. A Software Development Kit (“SDK”) can be downloaded onto a workstation to test the deployment of applications prior to use on the public internet. An introductory video on Google App Engine can be found here: <http://www.youtube.com/watch?v=bfgO-LXGpTM>

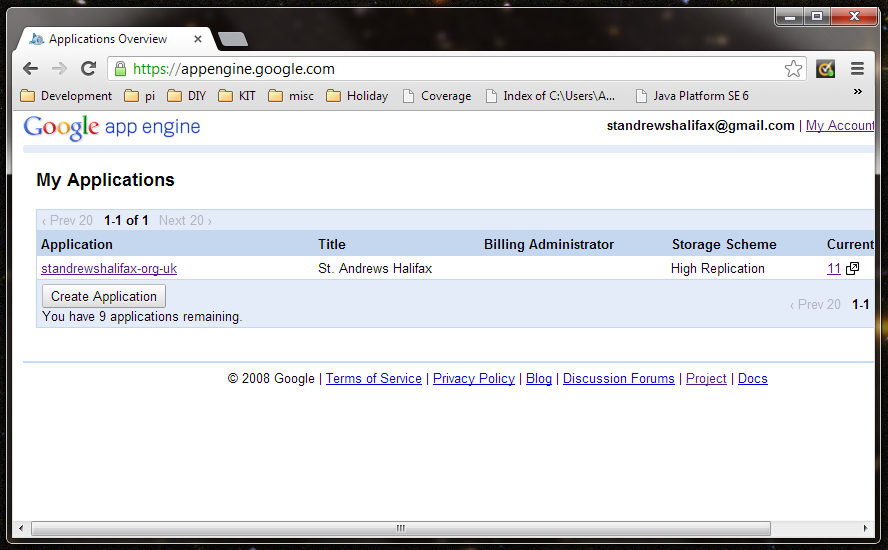
Google App Engine provides supporting services such as authentication, storage and caching. At present the St Andrews Halifax site only uses email. The Payer Vine and Visitors Form both use the email service to send a message to the Google account to [contact@standrewshalifax.org.uk](mailto:contact@standrewshalifax.org.uk).

Google App Engine is accessed using the Google Account: [standrewshalifax@gmail.com](mailto:standrewshalifax@gmail.com)

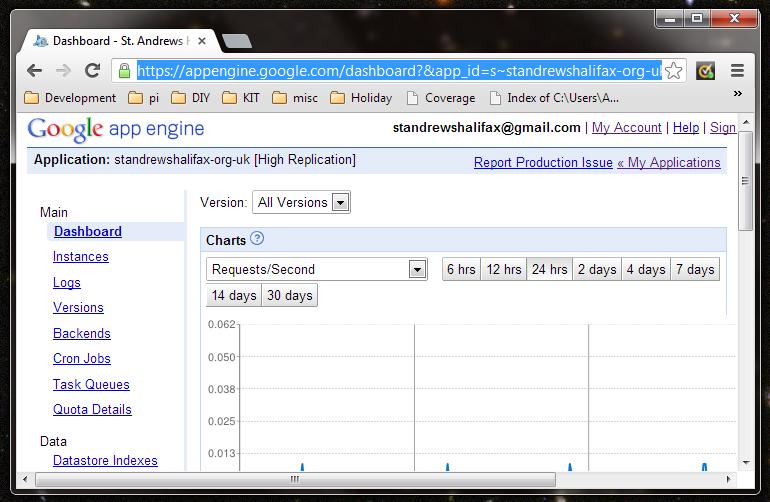
To access Google App Engine visit this URL: <http://appengine.google.com/>



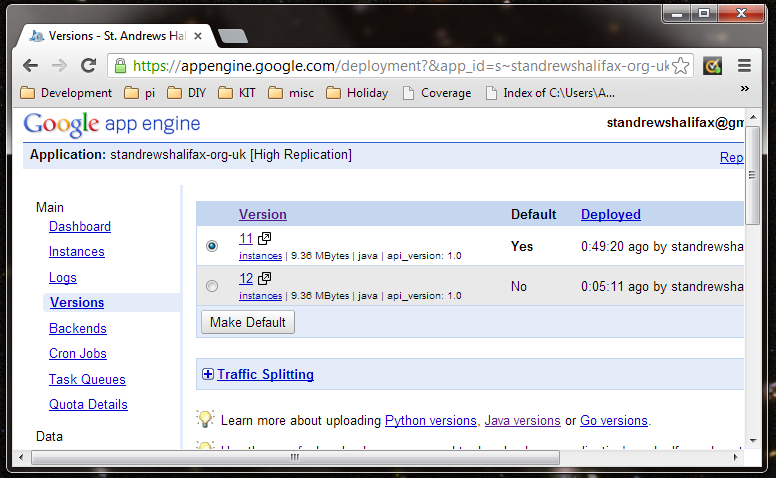
Sign in to see a list of applications:



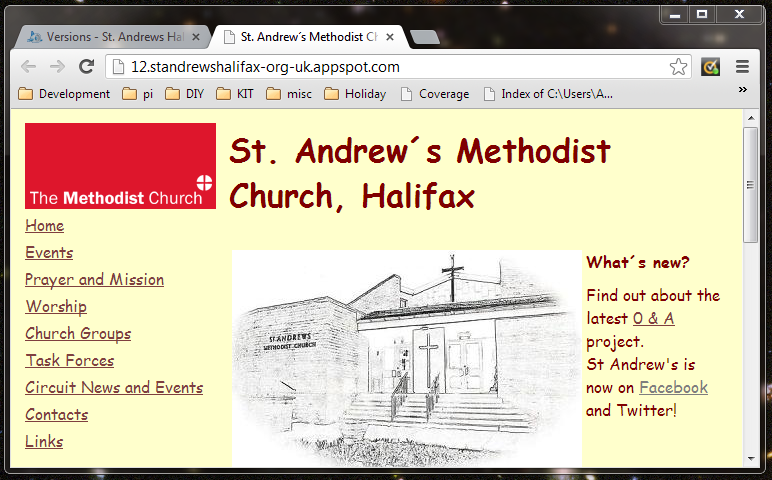
Click [standrewshalifax-org-uk](https://appengine.google.com/dashboard?&app_id=s~standrewshalifax-org-uk) to view the application dashboard:



Click Versions to see which versions have been uploaded to the site:

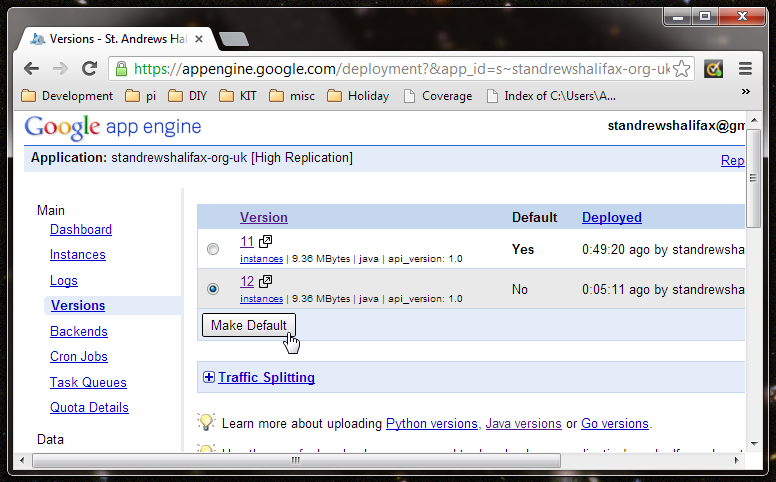


A specific version of the application can be viewed by clicking on the version number. Note the link in the browser contains the version number. In this case for version 12 we have: <http://12.standrewshalifax-org-uk.appspot.com/>



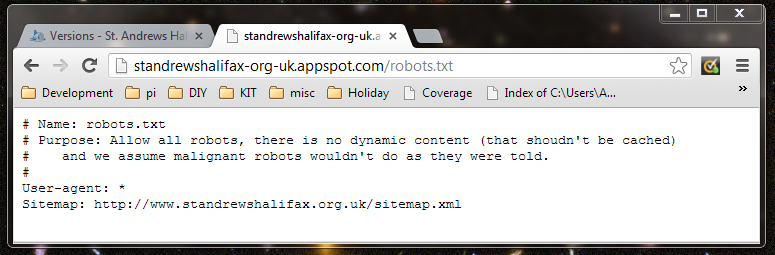
(Just after a new version has been uploaded it may take a few seconds for the site to load)

A different version can be enabled by selecting a version and clicking “Make Default”:

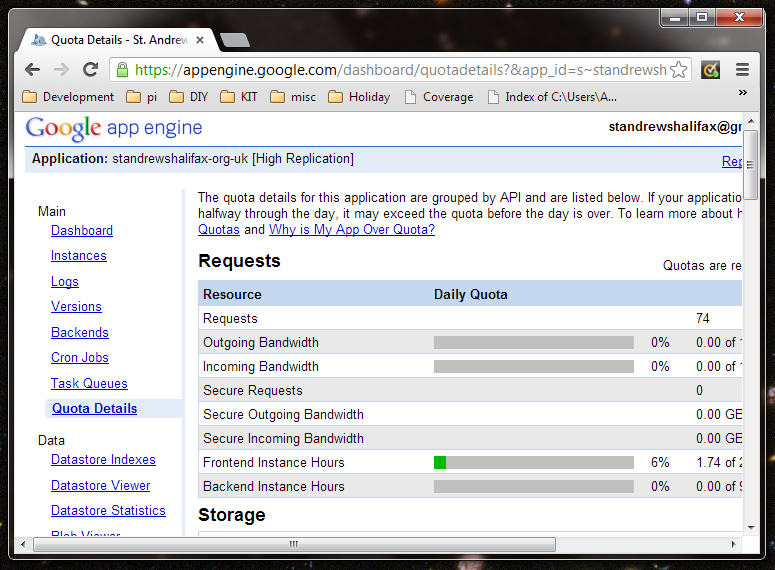


The default version is publicly available at <http://standrewshalifax-org-uk.appspot.com/> as well as the canonical URL of <http://www.standrewshalifax.org.uk/>. The .appspot.com version will be available even if the canonical URL of the site is not available. Any links given to external sites should only use the canonical URL to ensure consistency. To help search engines only to index the canonical URLs the site includes a robots.txt file reference to a sitemap at standrewshalifax-org-uk.

<http://standrewshalifax-org-uk.appspot.com/robots.txt> :



Click Quota Details to see how much of the daily usage quota has been consumed:

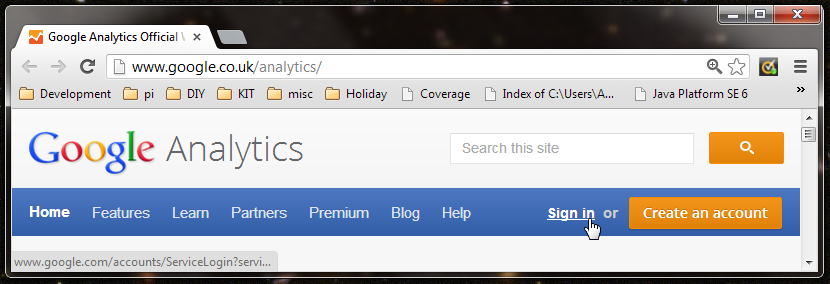


If the quota reached 100% in any day an additional allowance shall need to be purchased. Looking at the usage of St Andrews to date this is unlikely. In the event of a sudden spike in usage Google Analytics can be used to learn more about the source of the traffic.

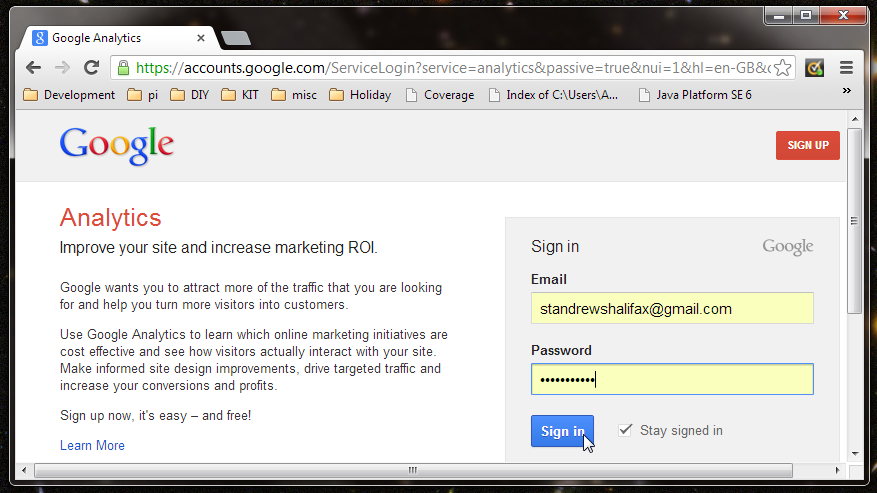
## Google Analytics

Google Analytics provides information on visitors to the site, where they came from and which pages they visited. This information supports decision making such as which pages are most popular and deserve more attention or which pages people aren’t finding and require increased visibility.

To access Google Analytics visit this URL: <http://www.google.co.uk/analytics/>



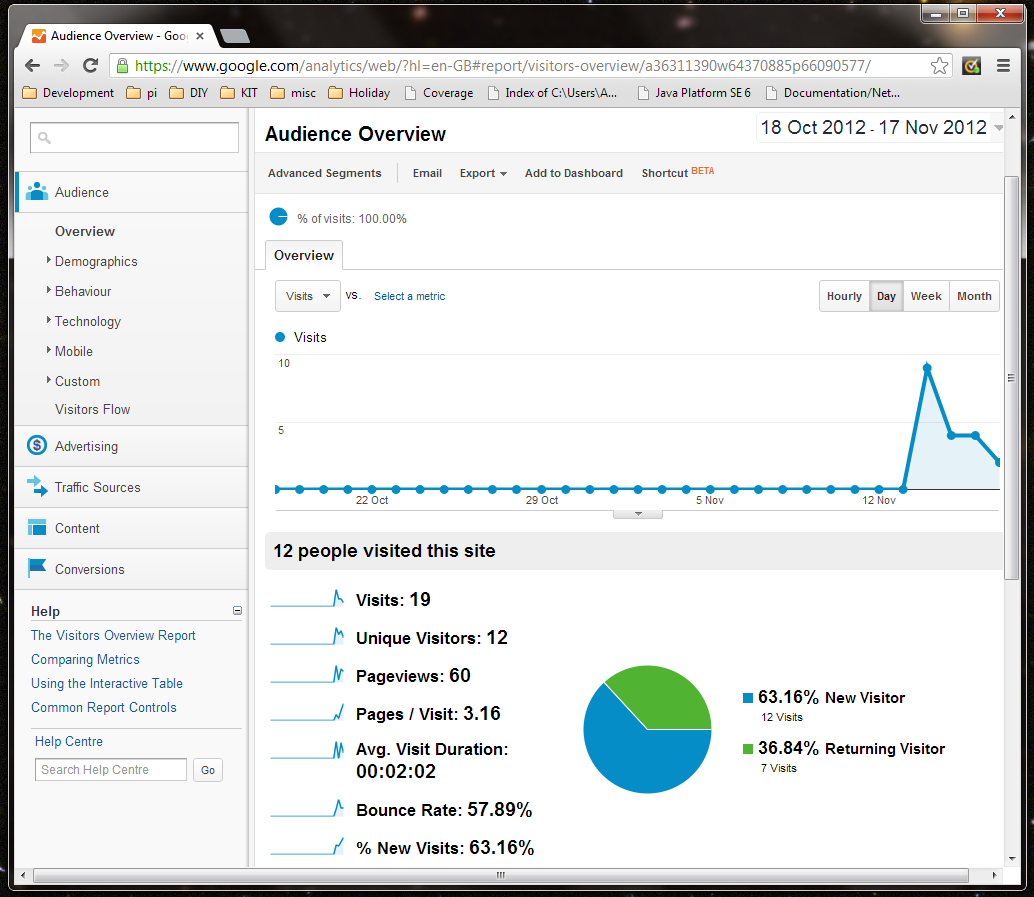
Click “Sign in” and sign in with the Google Account: [standrewshalifax@gmail.com](mailto:standrewshalifax@gmail.com)



Click the site URL the “All Web Site Data” to view the analytics:

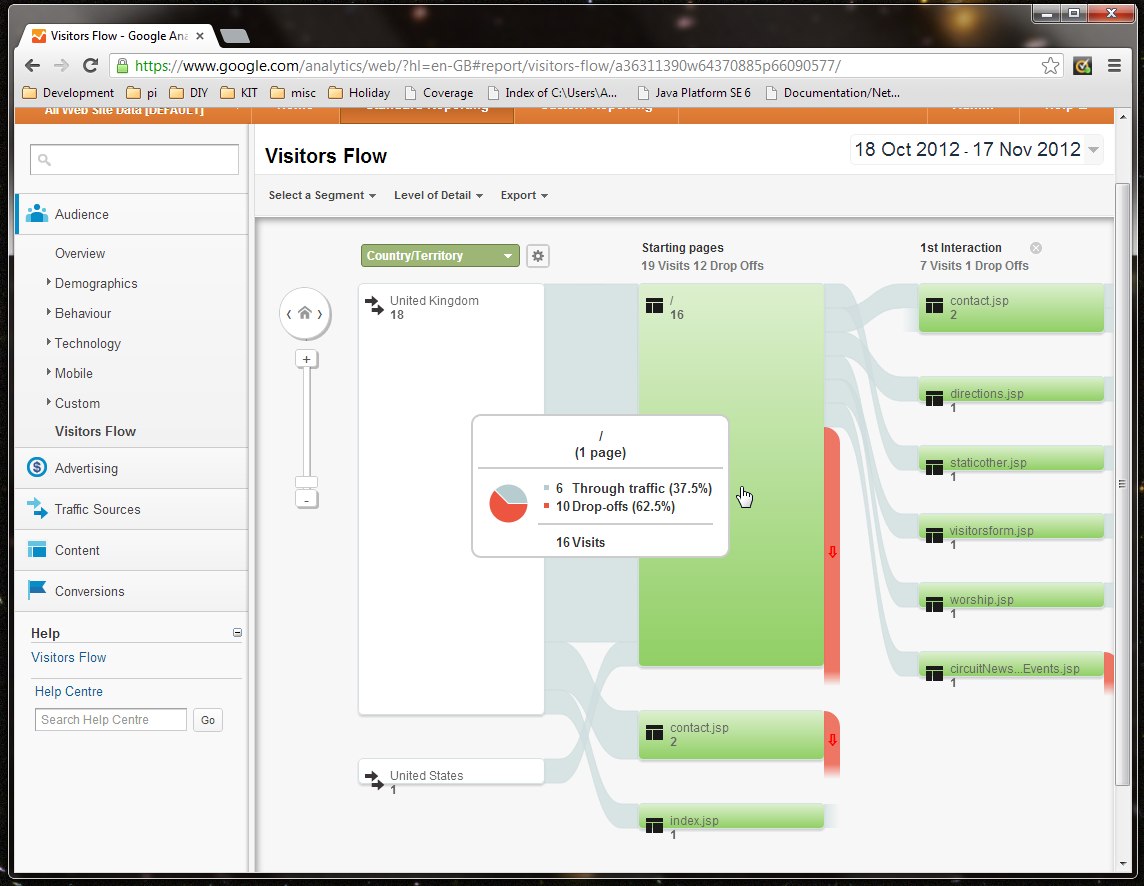


The analytics tool default to Audience Overview which has a traffic graph at the top and a breakdown of these visitors including a pie chart of new against existing visitors:



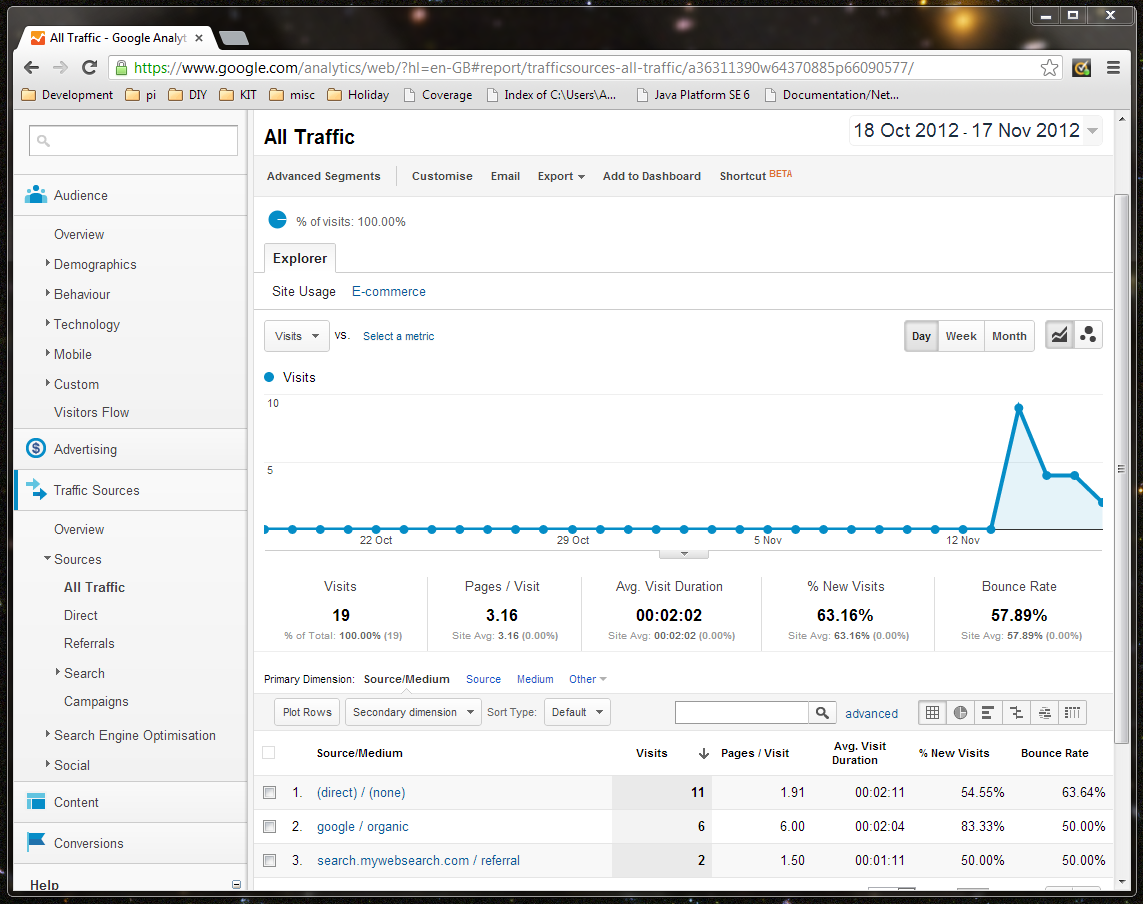
The graph is flat at 0 for the most of the last month as the Google Analytics tracking tags were added in the last few days.

The Visitors Flow analysis shows where visitors came from, which pages they visited and where they dropped off:



The drop off rate from the home page is 62% above. These type of visits are referred to as “bounces” meaning this visitor saw the home page but decided to look elsewhere for what they wanted.

Under “Traffic Sources”/“All Traffic” the visitors original location on the internet is shown.



Direct access is the most popular route to the site. This which means the site URL was entered directly in a browser. For a recently published site with relatively low volumes of traffic these results are likely to include lots of direct access during testing. “Google / organic” refers to visits to the site following a search using a search engine (in this case Google). Using Webmaster Tools we’ll be able to see what these search terms were.

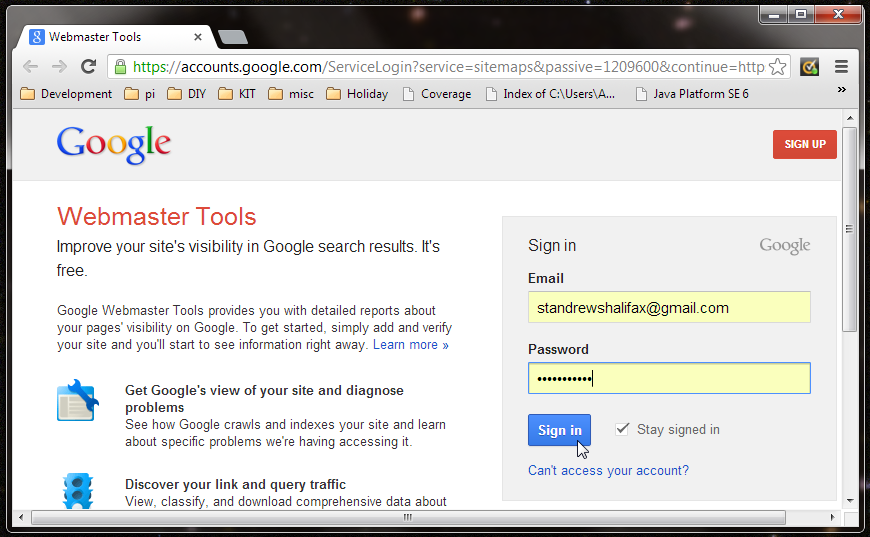
## Google Webmaster Tools

Google Webmaster Tools provide an insight into the performance and potential problems with the site. Problems like broken links will frustrate visitors and Google Webmaster Tools highlights these as well as other potential problems.

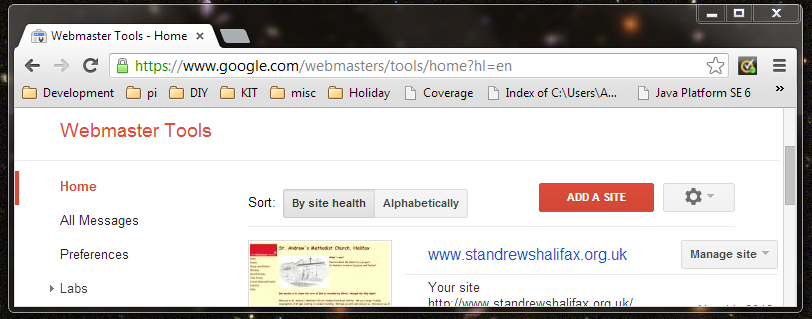
The search terms visitors use to access the site are also available. If a search phase implies visitors are looking for particular item but then bouncing this suggests that information is being picked up by the search engine but not by users, additional “sign posting” on the homepage may help.

Google Webmaster Tools is accessed using the Google Account: [standrewshalifax@gmail.com](mailto:standrewshalifax@gmail.com)

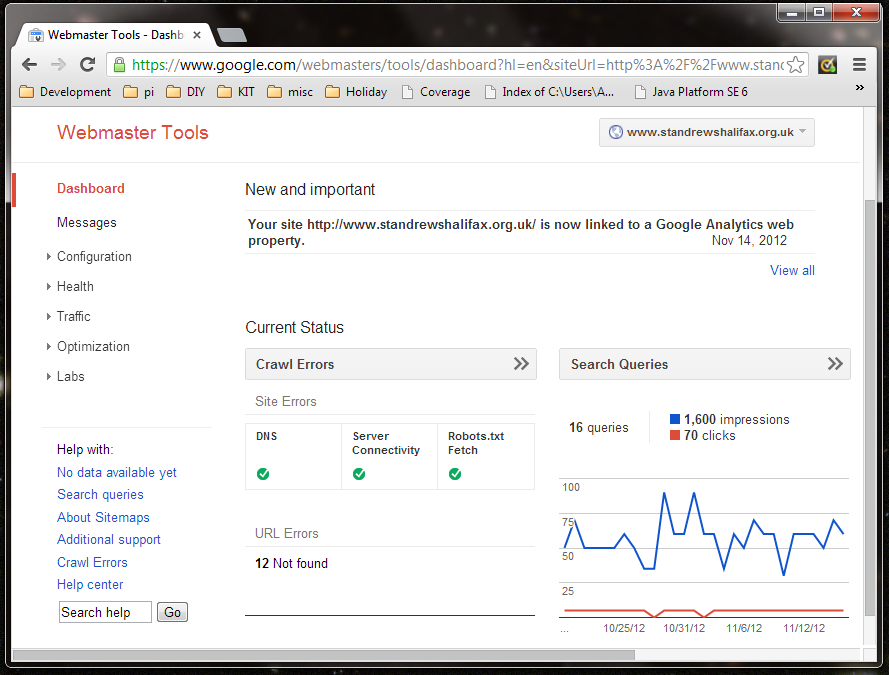
To access Google Webmaster Tools visit this URL: <https://www.google.com/webmasters/tools/>



Once signed in, any sites which are monitored through this account are listed:

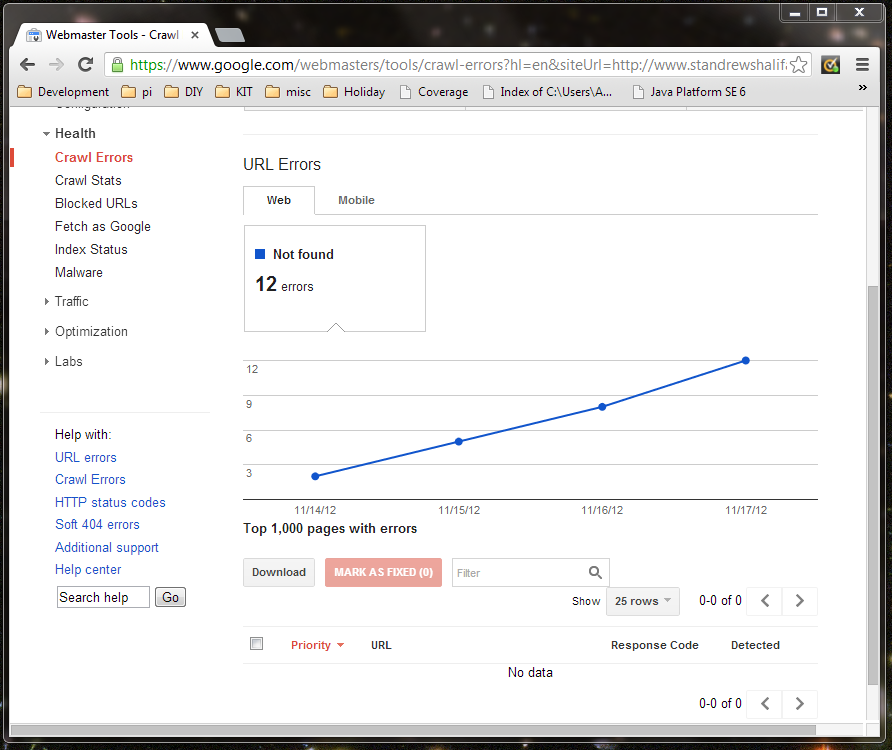


Click on the site URL [www.standrewshalifax.org.uk](https://www.google.com/webmasters/tools/dashboard?hl=en&siteUrl=http%3A%2F%2Fwww.standrewshalifax.org.uk%2F) to access the tools:



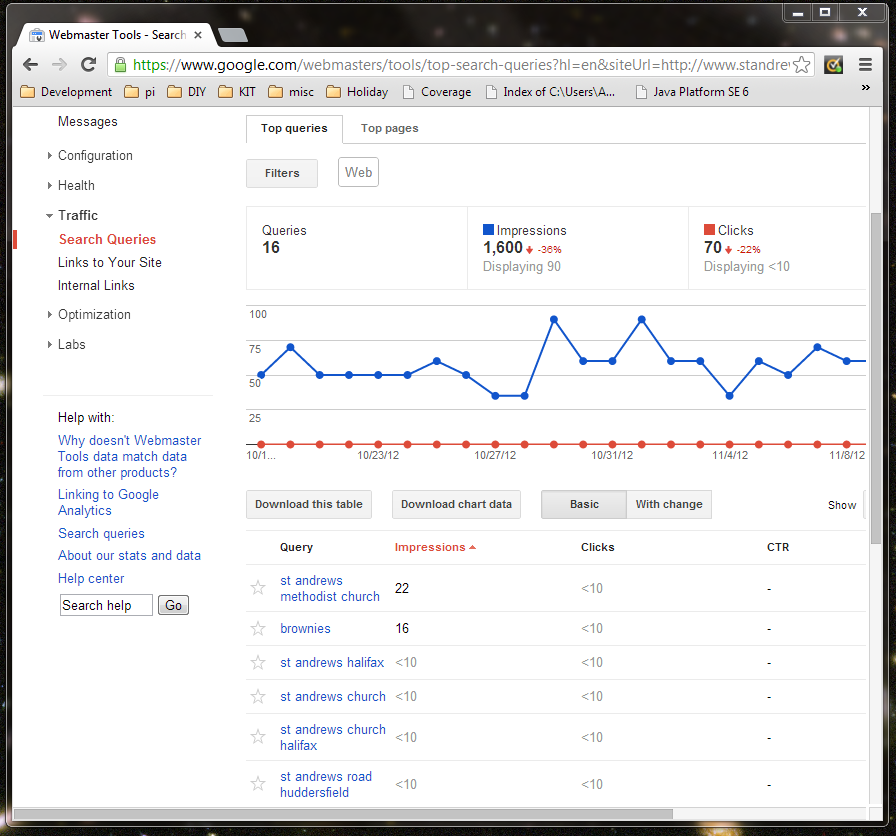
The comparison of clicks vs impressions is comparing the number of times the site appeared in search results (impressions) against the number of times the site was visited (clicks). These numbers do not correspond to visitor figures in Google Analytics as a full month’s search engine data is available from Google’s historic indexing of the site.

Select “Health”/“Crawl Errors” to see any errors Google Webmaster Tools has picked:



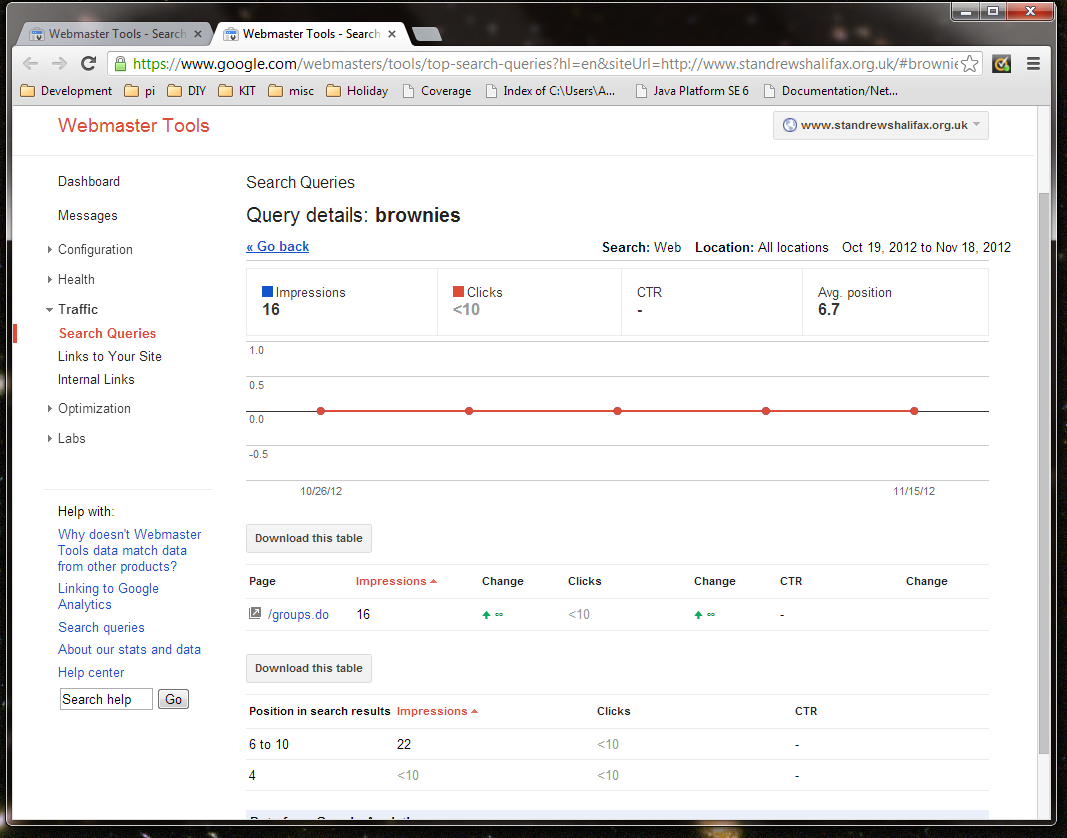
Here we see a recent increase in errors as some page links have been changed during the migration to Google App Engine but the search engine still had the old links indexed. All these have been marked as fixed but will re-appear if Google’s web crawler detects them again.

In “Traffic”/“Search Queries” the list of search terms used to discover the site is shown:



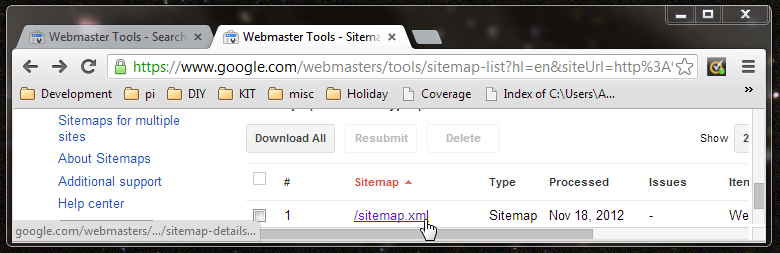
Here “brownies” is a surprise as it suggests the term is sufficient to find the St Andrews site. For such a widely used term, it would be reasonable to expect users to have passed by several other links before finding the St Andrews Site.

Click “brownies” to see where these visitors were navigating to:

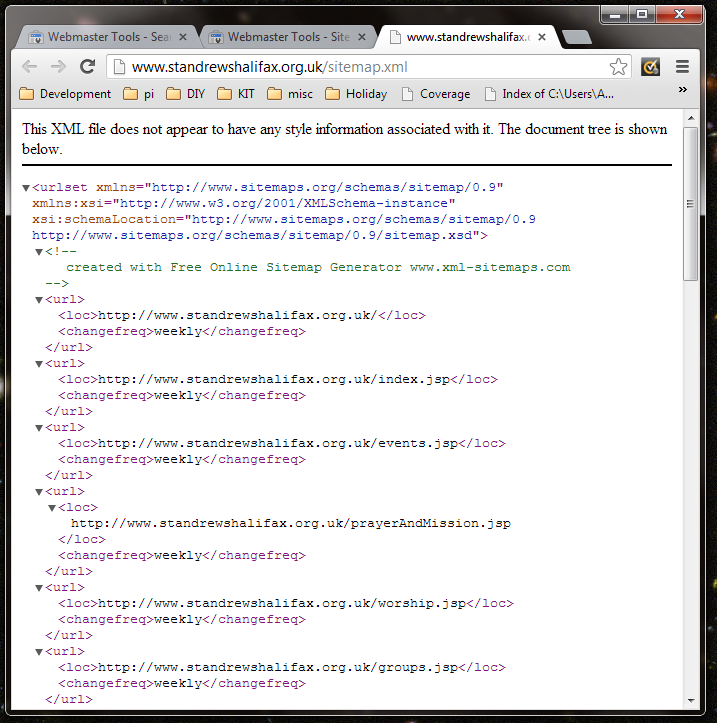


All 16 of the visitors who arrived after searching for “brownies” arrived at the groups page. Groups is shown as /groups.do which was the old path prior to the site being migrated to Google App Engine. This page is now: <http://www.standrewshalifax.org.uk/groups.jsp> .

To tell Google the pages address have changed a new sitemap can be submitted to get the page to be re-indexed. “Optimization”/“Sitemaps” shows the status of sitemap submissions:

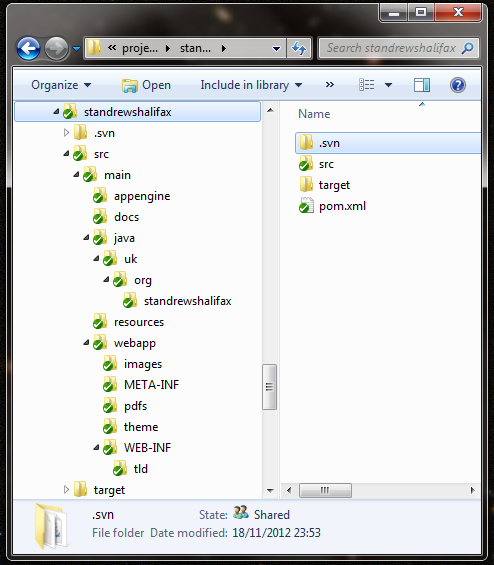


Bly clicking through the latest submission we can see what was submitted including the updated URL of the groups page



# Source structure

The website source is stored in a directory structure which is built into a single “Web Application”. The layout of this source is a standard Maven Web App and it looks like this:



More information on the Maven Standard Directory Layout can be found here: <http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>

## Folder Contents

The folders that contain files are broken down in the remainder of this section. Folders which are empty and have no significance on their own are omitted for clarity.

### standrewshalifax

The project root containing one file named pom.xml. This file controls how the web application is built. There are annotations in the file and an introduction to these files can be found here: <http://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

### standrewshalifax/.svn

.svn is a hidden folder containing version control information such as the original “pristine” copy of files that were checked out before they were edited. This folder and the files within it should not normally be manipulated directly. Further reading on Subversions system files can be found here: <http://subversion.apache.org/docs/release-notes/1.7.html#wc-ng>

### standrewshalifax/src

All source code, configuration and supporting files for the web application including any automated tests.

### standrewshalifax/src/docs

Supporting documents including this document. The docs folder is not built as part of the web application. The file standrewshalifax.org.uk.zone is a backup of the domain name entry for the site and can be used if the current settings are lost.

### standrewshalifax/src/main

For this application everything is under src/main, automated tests can be created and these would be under src/test but there are none at the time of writing.

### standrewshalifax/src/main/appengine

Configuration files specific to Google App Engine. Just one file: appengine-web.xml with most values pre-populated from the pom.xml. The file is copied below and the replaced values are prefixed with $ and wrapped in {curly braces}.

<?xml version="1.0" encoding="utf-8"?>

<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">

<application>${project.artifactId}</application>

<threadsafe>true</threadsafe>

<version>${project.version}</version>

<precompilation-enabled>false</precompilation-enabled>

</appengine-web-app>

### standrewshalifax/src/main/java

Java application source files which compile into class files. A Java namespace convention is that files should be stored in a directory structure matching the domain name associated with their ownership. Under here you see a sub directory “uk”, under that “org” and under that “standrewshalifax”. As St Andrews owns standrewshalifax.org.uk no other organisation should use that structure.

### standrewshalifax/src/main/java/uk/org/standrewshalifax

The Java source files which provide logging, email dispatch and a utility class which strips the surrounding header and footer information from the HTML fragments which are then included in other pages.

* Constants.java - Reusable values usd throughout the application.
* JavaMailHelper.java – A wrapper for the Java mail library.
* LogHelper.java – A logging utility which dumps system information for debugging.
* MailDispatcher.java – A simplified interface for pages to invoke email sending.
* Utilities.java - Utilities for loading page fragments.

Recent updates have trimmed down these classes to the extent that most contain only 1 or 2 functions. Some further rationalisation may be able to take place to consolidate all functions unto a single class.

### standrewshalifax/src/main/resources

The “resources” are miscellaneous files available to the application. In this case there are 2 system resources; logging.properties which controls the level of detail output in the logs and version.properties which is automatically populated with version information. The remaining files are HTML fragments which can be manually edited and included with the web application. These files are:

* BacktoChurchSunday.xhtml
* alpha.xhtml
* aol.xhtml
* beachparty.xhtml
* clowningaround.xhtml
* essence.xhtml
* fasta.xhtml
* letter.xhtml
* outreachandawareness.xhtml
* prayerandmission.xhtml
* scouts.xhtml
* whatsnew.xhtml
* yw.xhtml

### standrewshalifax/src/main/webapp

Files in this folder are visible to the public under the root of the web application. For example robots.txt (a hint for search engines) is in the webapp folder and is available here: <http://www.standrewshalifax.org.uk/robots.txt>

Some files are structural and include all the content from either constant files or from other JSP files. An example of this in index.jsp which contains the following key section:

<tiles:insert template="/simpleTemplate.jsp">

<tiles:put name="title" content="<%=Constants.SITE\_TITLE%>" direct="true" />

<tiles:put name="titlelink" content="<%=Constants.PAGES[Constants.INDEX\_ID][Constants.ACTION]%>" direct="true" />

<tiles:put name="subtitle" content="<%=Constants.PAGES[pageId][Constants.NAVNAME]%>" direct="true" />

<tiles:put name="header" content="titleHeaderTile.jsp" />

<tiles:put name="left" content="navigationLeftTile.jsp" />

<tiles:put name="body" content="<%=Constants.PAGES[pageId][Constants.BODY]%>" />

</tiles:insert>

Key information here is the template (simpleTemplate.jsp) which contains the overall page structure and the reference to the body fragment: Constants.PAGES[pageId][Constants.BODY]. The body fragment file can be found by looking up the reference in the constant file. This indirection is overkill and is may be prudent to clean these up using direct references to files instead of the constants. The convention used in most places if for a layout file called name.jsp to be accompanies by a body fragment called nameBody.jsp.

### standrewshalifax/src/main/webapp/images

The images folder contains image files used throughout the site and these are publicly available for example: <http://www.standrewshalifax.org.uk/images/clowningaround.gif>

### standrewshalifax/src/main/webapp/META-INF

There is one file in this folder: context.xml . This is a system file which is not publicly available and contains no information but is required by the more recent versions of the Tomcat servlet container to run a web application.

### standrewshalifax/src/main/webapp/pdfs

PDF files are collected in this folder and linked from other pages in the site. Some of these have been there for some time including <http://www.standrewshalifax.org.uk/pdfs/Beach_Party_photos.pdf> which is 5 and a half years old.

### standrewshalifax/src/main/webapp/WEB-INF

The Web-INF folder is a system folder containing definitions for some page structure utilities such as <tiles:insert> as seen above. The web.xml file contains the welcome file list which is the names of files the application attempts to find when a user visit’s the root of the web application. There is just one file named in the list (index.jsp) which is rendered as the home page.

### standrewshalifax/target

The target folder contains compiled application code. When the site is deployed to Google App Engine files from this folder are uploaded. When the web application source is check out from Subversion, this folder is not present as it is built from the contents of src/main.

# Development

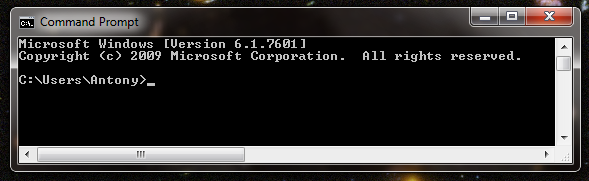
## System requirements

To complete the instructions below an internet connected computer with a Windows, Mac or Linux operating system installed. Each individual software package will have its own specific hardware and software requirements. A modestly priced PC purchased in the last 3 years with a domestic broadband connection should be sufficient.

The instructions below were created using Windows 7 and have not been tested on other operating systems. To install software you shall need a user with administrative privileges. To complete this section installing software is something you should already be comfortable with and you should be able to unzip a file.

Using the command prompt is a new skill you may be learning. When asked “to open a Command Prompt” the following is expected:

1. Press the Windows start button
2. Type “Command Prompt” into the program search box (the cursor is be there already)
3. Press return
4. A window list this will appear:



## Developer Workstation set-up

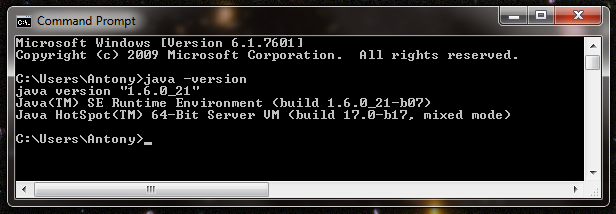
This section includes several software components that shall need to be installed. In most cases the installation is automated but where this is not the case install steps have been captured from windows 7.

The applications which are installed are not specific to the St Andrews Website (apart from the source itself) and it is possible to create your own web applications using the same tools.

### Install Java & check

Java is the language the website is written in. Even though there is little Java code to edit the web servers and build tools are built from Java and require it to run.

1. Visit the Java Platform Standard Edition Development Kit download page here: <http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u26-download-400750.html>
2. Select “Accept License Agreement” and select the most appropriate version of the software to install. More recent computers will be running Windows x64 (64 bit).
3. If you are not sure which version of windows you have, press the Start button, type “System Information” in the search box and press return. The type of operating system is next to “System Type”.
4. Install Java following the instructions as prompted.
5. Once Java is installed confirm this is the case by opening the Command Prompt and typing “java -version”, then press return. You should see the Java version message as below:



1. Close the Command Prompt

### Install Google app engine SDK

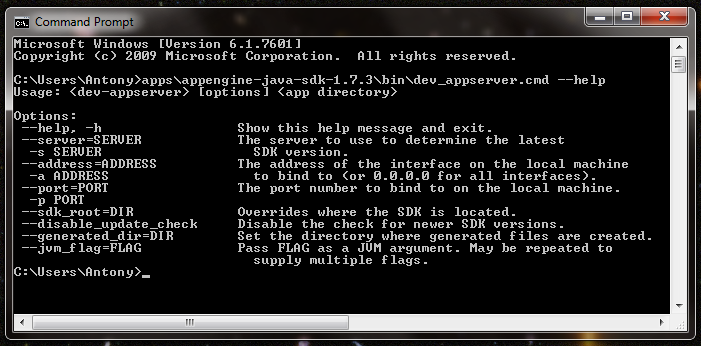
Google App Engine includes a web server to run the application locally and relies on the Java installation just completed.

1. Create a folder called “apps” under your home directory. This is the folder the Command Prompt opened in by default above.
2. Visit the Google App Engine download page here: <https://developers.google.com/appengine/downloads?#Google_App_Engine_SDK_for_Java>
3. Download the “Google App Engine for Java”
4. Unzip appengine-java-sdk-1.7.3.zip into the newly created apps folder so the following path exists: <your home folder>\apps\appengine-java-sdk-1.7.3\bin

(Replace x and y with the App Engine SDK versions.)

1. Opening the Command Prompt and type:

“apps\appengine-java-sdk-1.7.3\bin\dev\_appserver.cmd --help” (there are 2 minus signs), then press return. You should see the Java version message as below:



1. Close the Command Prompt

### Install SVN client

### Install Maven & check

### Install Notepad ++

### Check out code

### Build application

### Run in local SDK

### Check with browser

### 

## Updates and publication

### Update your workspace

### Check version number on deployed version

### If the current number is in use, move to next number

### Make changes

### Build using mvn clean install

### Deploy to local version and check

### Deploy to remove version and check

### Switch default version to the new version

### Commit changes