

# Table of Contents

<b>Django Site Development</b> .....	3
<b>Django documentation</b> .....	3
<b>File layout</b> .....	3
<b>Template dependencies</b> .....	4
<b>Applications installed</b> .....	5
<b>Sites and directories</b> .....	6
<i>http://oclug.on.ca</i> .....	6
<i>http://staging.oclug.on.ca</i> .....	7
<i>http://sandbox.oclug.on.ca</i> .....	7
<b>Scripts</b> .....	8
<i>nightly</i> .....	8
<i>serialize_db_and_commit</i> .....	8
<i>reset_sandbox</i> .....	8
<i>update_staging</i> .....	9
<i>push_staging_to_production</i> .....	9
<b>Open issues</b> .....	9
<b>Resolved issues</b> .....	9



[\[PageOutline\]](#)

# Django Site Development

- RemoteDjangoSiteDevelopment - SVN repository info etc.

The new Django site code creates two types of site administrators where before there was only one:

1. People updating Meetings, Announcements, Page content
2. People adding new features to the site

The former group can now do everything they need to do through the admin interface built in to the site at: <http://oclug.on.ca/admin/>

The latter group needs a way to work on the code collaboratively, test changes and update the database schema when the changes are deployed.

## Django documentation

Django has very good [online documentation](#). The [tutorial](#) is a good place to start. Some other useful references:

- [Models](#)
- [URLs](#)
- [Templates](#)

## File layout

Shortcuts to some of the files available from the “browse source” button above.

file	description
<a href="#">__init__.py</a>	allows directory to be imported as a module (python idiom)
<a href="#">local_settings.py</a>	database and secret key settings NOT IN SUBVERSION
<a href="#">settings.py</a>	global settings for the project
<a href="#">urls.py</a>	static mappings for URLs to views and templates
<a href="#">sandbox_settings.py</a>	overrides for sandbox site
<a href="#">staging_settings.py</a>	overrides for staging site
<a href="#">manage.py</a>	django utility script for managing the project
<a href="#">dump.sh</a>	script to dump all the non-sensitive data from the db as xml
<a href="#">docroot</a>	<b>apache document root</b>

<a href="#">docroot:favicon.ico</a>	special case fall-through for “favorites icon”
<a href="#">docroot:media:</a>	all requests to <a href="#">oclug.on.ca/media/</a> are served from this directory (apache config)
<a href="#">docroot:media:admin</a>	symlink to django's admin site media directory (for icons, etc.)
<a href="#">docroot:media:uploaded</a>	symlink to <code>site_data/uploaded</code> directory (for files and images uploaded to admin site)
<a href="#">docroot:media:oclug.css</a>	CSS for site
<a href="#">docroot:media:oclug.png</a>	site logo
<a href="#">meetings</a>	<b>custom OCLUG meetings application</b>
<a href="#">meetings:models.py</a>	meetings models, database field types and admin site customization
<a href="#">meetings:views.py</a>	custom view functions for meetings application
<a href="#">meetings:feeds.py</a>	RSS feed classes for meetings
<a href="#">meetings:templatetags</a>	extension code for templates
<a href="#">meetings:templatetags:meetings_image.py</a>	templatetag for uploaded images (to include size, title, alt, etc.)
<a href="#">templates</a>	<b>templates that generate site HTML</b>
<a href="#">templates:base_generic.html</a>	common site look, with logo, navigation bar and boxes
<a href="#">templates:404.html</a>	404 (not found) page
<a href="#">templates:500.html</a>	500 (server error) page
<a href="#">templates:meetings</a>	<b>templates for meetings application</b>
<a href="#">templates:meetings:upcoming.html</a>	page for upcoming meetings and announcements
<a href="#">templates:meetings:past.html</a>	page past meetings
<a href="#">templates:meetings:meeting_detail.html</a>	page for an individual meeting
<a href="#">templates:meetings:location_detail.html</a>	page for a meeting location
<a href="#">templates:meetings:speaker_detail.html</a>	page for a speaker
<a href="#">templates:meetings:image_detail.html</a>	page for single-image view
<a href="#">templates:meetings:meeting_brief.html</a>	a one-line view of a meeting (page fragment)
<a href="#">templates:meetings:meeting_info.html</a>	a detailed view of a meeting (page fragment)
<a href="#">templates:admin</a>	<b>overrides for admin site templates</b>
<a href="#">templates:admin:base_site.html</a>	modified version that includes a link to markdown reference
<a href="#">templates:feeds</a>	<b>RSS feed templates</b>
<a href="#">templates:feeds:upcoming_description.html</a>	meeting description (currently blank)
<a href="#">templates:feeds:upcoming_title.html</a>	meeting title
<a href="#">templates:flatpages</a>	<b>flatpages application templates</b>
<a href="#">templates:flatpages:default.html</a>	flatpage layout

## Template dependencies

```
#!rst
```

```
+-----+-----+
-----+
```

```

| "extends"                |page template                | "includes"
|
+=====+
=====+
|base_generic.html        |meetings/upcoming.html
|meetings/meeting_info.html |
+-----+
+
|                          |meetings/meeting_detail.html |
|
+-----+
-----+
|                          |meetings/past.html
|meetings/meeting_brief.html |
+-----+
+
|                          |meetings/speaker_detail.html |
|
+-----+
+
|                          |meetings/location_detail.html|
|
+-----+
-----+
|                          |meetings/image_detail.html  |
|
+-----+
+
|                          |flatpages/default.html     |
|
+-----+
+
|                          |400.html                    |
|
+-----+
+
|                          |500.html                    |
|
+-----+
-----+

```

# Applications installed

```

#!rst
=====

```

```

=====
application                description
-----
-----
django.contrib.auth        users, groups and permissions
django.contrib.contenttypes
django.contrib.sessions    session tracking (required by auth)
django.contrib.sites       multiple site support (required by flatpages)
django.contrib.admin       administration interface
http://oclug.on.ca/admin/
django.contrib.markup      markup template tags, used for markdown
formatted text
django.contrib.flatpages   database-backed pages, used for
http://oclug.on.ca/about/, etc.
django.contrib.redirects   database-backed redirection, used to fix .php
urls mostly
meetings                   custom OCLUG meetings application
=====
=====

```

## Sites and directories

### <http://oclug.on.ca>

Main OCLUG site

```

#!rst
=====
=====
code location      /home/webdude/oclug_django_site/
"push_staging_to_production" script used to copy from staging
settings file     settings.py
data location     /home/webdude/site_data/          checked in to SVN:
/site_data/trunk/
upload location   /var/local/main_site_uploaded/    owned by www-data,
file and image uploads go here
database         Postgres: "oclug_django_site"
serialized database /home/webdude/site_data/db.xml
"serialize_db_and_checkin" script called nightly
=====
=====

```

## http://staging.oclug.on.ca

Site for testing new code and db columns/tables

```

#!rst
=====
=====
code location      /home/webdude/staging/oclug_django_site/ "update_staging"
script used to update from SVN
settings file      staging_settings.py
data location      /home/webdude/site_data/                shared with main
site
upload location    /var/local/main_site_uploaded/          shared with main
site
database           Postgres: "oclug_django_site"            shared with main
site
serialized database none                  data not
serialized
=====
=====

```

## http://sandbox.oclug.on.ca

Site for testing changes made by admin interface

```

#!rst
=====
=====
code location      /home/webdude/sandbox/oclug_django_site/ copied from main
site by "reset_sandbox" script
settings file      sandbox_settings.py
data location      /home/webdude/sandbox/site_data/        copied from main
site by "reset_sandbox" script
upload location    /var/local/sandbox_site_uploaded/       copied from main
site by "reset_sandbox" script
database           Postgres: "oclug_django_sandbox_site"    reloaded from
main site by "reset_sandbox" script
serialized database none                  data not
serialized
=====
=====

```

main and staging sites share database and upload directory so that common changes (adding columns to a table and creating new models) can be tested with the live data. When there are new columns or tables

the main site will ignore them.

## Scripts

This is a list of the maintenance script locations and what they do.

### nightly

/home/webdude/bin/nightly (run from crontab daily at 4am)

1. call `serialize_db_and_commit`
2. call `reset_sandbox`

### serialize\_db\_and\_commit

/home/webdude/bin/serialize\_db\_and\_commit (called by nightly script)

1. dump the database, except sessions and user auth info, into the serialized database location
2. rsync uploads into site data dir

3. check in everything in site data dir to SVN. This will cause changes to be emailed out like other SVN commits, and everyone on the `oclug-www` list will be able to see what has changed on the site.

### reset\_sandbox

/usr/local/bin/reset\_sandbox → /home/webdude/bin/reset\_sandbox (called by nightly script)

1. rsync `oclug_django_site` and `site_data` from main site to sandbox (see `/root/bin/rsync_oclug_sandbox`)
2. dump site database into sandbox database
3. restart apache

## update\_staging

/usr/local/bin/update\_staging → /home/webdude/bin/update\_staging

1. “svn update” staging oclug\_django\_site dir
2. “manage.py validate && manage.py syncdb” if any errors, stop and notify user

3. restart apache

## push\_staging\_to\_production

/usr/local/bin/push\_staging\_to\_production → /home/webdude/bin/push\_staging\_to\_production

1. rsync oclug\_django\_site from staging
2. restart apache

## Open issues

How do we accommodate changes to the database layout (other than new tables that are handled by “manage.py syncdb”)?

- new columns can be dealt with by manually entering “ALTER TABLE” commands (not as hard as it sounds, can use “manage.py sql ...” to show the sql that would be used to create the table and just copy&paste the column into “psql”)
- need to investigate serialize/flush/syncdb/reload as a possible option (maybe not, some things shouldn't be made too simple)

## Resolved issues

- need to figure out why subversion notifications aren't being sent - [logfile20070626](#)
- install Trac for issue tracking and development - [logfile20070712](#) [logfile20070717](#)

From:

<https://wiki.linux-ottawa.org/> - **Linux-Ottawa (OCLUG) Wiki**

Permanent link:

<https://wiki.linux-ottawa.org/doku.php?id=djangositedevelopment>

Last update: **2015/06/09 19:23**

