

# Table of Contents

<b>gLite Web Pages.....</b>	<b>1</b>
Introduction.....	1
Application Structure.....	1
Feed the database from the release scripts.....	1
savannah_to_web.py.....	1
test_preview_to_production_to_web.py.....	1
Modify and publish a release.....	2
Install the web pages.....	2

# gLite Web Pages

## Introduction

The new version of the gLite webpages is dynamic and provides a solution to the problems of the previous static pages. It's written in Django with all the release information stored into a database. And there is also an admin interface to introduce or modify the release information.

The web application it's integrated with the new release scripts to feed the database at the same time the repository it's created. Therefore, it's not needed to perform an installation to retrieve the information as it was done before.

The node used for creating the repositories and for hosting the web pages is lxbrf2713.

You can find the release scripts and the web application source code in <https://svn.cern.ch/repos/grelscripts/>

## Application Structure

TODO

## Feed the database from the release scripts

There are two scripts in the repository scripts to feed the release information to the web pages: `savannah_to_web.py` and `test_preview_to_production_to_web.py`

### **savannah\_to\_web.py**

It can be used to insert the information of the patches that are going to be released into the web page database. This script will take the information from a patch (or many patches) from savannah, it will transform the information to the format needed by the web page, and it will insert or update the patch into the database. This script can be launched as many times as needed, but be careful because any manual changes will be overwritten.

The patch insertion can fail for many reasons, but unless it's specified in the error code, usually it's because the release, platform, service or service platform are not defined properly for this patch. For example, if we introduce a new service, and we forget to create the service and service platform before using this script it will fail.

After inserting a patch, it's usually required to perform some format or style modifications, but that will be explained afterwards.

Please, note also that this script doesn't insert all the information needed for the release. In particular, the rpms that have been updated and their URLs are missing after inserting a patch.

### **test\_preview\_to\_production\_to\_web.py**

This script it's used to provide the missing information about the updated rpms and their URLs. It's execute in the same way as `preview_to_production.py` and it does exactly the same steps, but without actually touching the prepare repos. **YOU HAVE TO USE IT BEFORE `preview_to_production.py` !!!** Otherwise, you will need to clean prepare/prod, and copy a clean version from production.

After the staged rollout, when the beta (preview) repository has been test, you can run this script to detect the rpms that have been updated since the last release, and to generate the URLs that will have in the production repository. This information will update the patch in the web pages database to include this information, but won't touch any of the other fields in the patch. Therefore, it can be safely used after modifying the patch by hand, for example to correct the release notes format.

## Modify and publish a release

For detailed information about the release process please check the Staged Rollout procedure and the Production procedure.

## Install the web pages

- An SL5 machine it's recommended
- Enable the EPEL repo (needed for mod\_wsgi) modifying /etc/yum.repos.d/epel.repo
- Install apache2 and mysql
  - ◆ Install mysql: yum install mysql-server
  - ◆ Install apache2: yum install httpd mod\_ssl mod\_wsgi
- Download the source code from <https://svn.cern.ch/repos/grelscripts/>
  - ◆ cd /srv && svn co <https://svn.cern.ch/repos/grelscripts/> --username your\_username\_here
- Link grelscripts/glite\_web from /srv and make sure apache can read it.
  - ◆ ln -s grelscripts/glite\_web/ && chown -R apache:apache glite\_web
  - ◆ chown -R apache:apache grelscripts/glite\_web
  - ◆ mkdir /var/www/forbiden && chown -R apache:apache /var/www/forbiden
- Copy the apache configuration and configure it
  - ◆ cp -b glite\_web/httpd.conf /etc/httpd/conf/
  - ◆ Change the domain name to the new one (if that's the case) in /etc/httpd/conf/httpd.conf
  - ◆ Uncomment the last line in /etc/httpd/conf.d/wsgi.conf to enable the module
  - ◆ Comment all the lines in the Virtualhost at the end of the file /etc/httpd/conf.d/ssl.conf
- Install django 1.1
  - ◆ Install dependencies: yum install python-simplejson
  - ◆ For example, you can use this rpm: rpm -i <http://koji.afroditi.hellasgrid.gr/packages/Django/1.1.1/2.el5/noarch/Django-1.1.1-2.el5.noarch.rpm> or take it from [http://quattorsrv.lal.in2p3.fr/packages/egee-SA1/release/sl5/x86\\_64/Django-1.1.1-2.el5.noarch.rpm](http://quattorsrv.lal.in2p3.fr/packages/egee-SA1/release/sl5/x86_64/Django-1.1.1-2.el5.noarch.rpm)
- Install python mysql module (version 1.2.1\_p2)
  - ◆ Install dependencies: yum install mx
  - ◆ For example you can install it from: rpm -i [http://quattorsrv.lal.in2p3.fr/packages/egee-SA1/release/sl5/x86\\_64/MySQL-python-1.2.1\\_p2-1.el5.x86\\_64.rpm](http://quattorsrv.lal.in2p3.fr/packages/egee-SA1/release/sl5/x86_64/MySQL-python-1.2.1_p2-1.el5.x86_64.rpm)
- Install South database migration tool:
  - ◆ wget <http://www.aeracode.org/releases/south/south-0.7.2.tar.gz>
  - ◆ tar -xvzf south-0.7.2.tar.gz
  - ◆ cd south && python setup.py install
- Start apache and mysql
  - ◆ /etc/init.d/mysqld start
  - ◆ /etc/init.d/httpd start
- Migrate the database
  - ◆ Enter in mysql and execute: mysql
  - ◆ Create the database: "create database glite\_web char set utf8;"
  - ◆ Give permissions to a user in the database: "grant all on glite\_web.\* to glite@localhost identified by 'thepassword';"
  - ◆ Exit form mysql: exit
  - ◆ Load a backup: mysql glite\_web <

- /afs/cern.ch/project/gd/gliteweb-backup/backaup\_file\_to\_restore
  - ◆ Change username and password in glite\_web/settings.py
  - ◆ Update the schema: glite\_web/manage.py migrate
  - Change the secret key in grelscripts/glite\_web/settings.py ( you can use <http://mylesbraithwaite.com/journal/2007/10/secret-key/> )
  - Restart the web server: /etc/init.d/httpd restart
  - Configure autostart of afs, apache and mysql so that it will start again after a reboot (it's what you want):
    - ◆ chkconfig --levels 345 afs on
    - ◆ chkconfig --levels 345 mysqld on
    - ◆ chkconfig --levels 345 httpd on
- 

This topic: LCG > GliteWebPages

Topic revision: r5 - 2011-01-20 - unknown



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)