

# Graphviz Website Installation, Administration and Maintenance

## 1 Overview

The graphviz.org website is based on the Drupal content management system. Drupal uses a MySQL database to store web pages and information about users such as user id, password and permissions. Website configuration details are placed in the database as well. This document describes file locations, installation steps, website administration, user operations and changes to the build process. It serves as a reference that can be used to understand, maintain and trouble shoot the website.

## 2 Supporting Systems

Drupal websites rely on supporting systems including PHP, Apache web server and MySQL database management system. The following software versions are installed and used by the graphviz.org website:

- php-5.1.6-27.el5\_5.3 (rpms install)
- Apache/2.2.3 (CentOS-5.5)
- mysql-5.0.77-4.el5\_5.4 (rpms install)
- drupal-6.20 (drupal-6.20.tar.gz)

## 3 File Locations

Files used by the graphviz.org website include those associated with the supporting systems and those associated with web content. The rpms installations placed files in standard locations. There are a few key file locations that are of interest.

- PHP - the php configuration file is located at /etc/php.ini. This file has not been changed.
- MySQL
  - the website database files are located at /var/lib/mysql
  - the MySQL configuration file is located at /etc/my.cnf
  - the error log file is located at /var/log/mysqld.log
- Apache
  - the **document root directory** for the port 8080 server is /home/dperry/myhttpd/html
  - the Apache configuration file is located at /home/dperry/myhttpd/conf/httpd.conf
- Drupal - the files that enable Drupal CMS are located in the **document root directory**
- Website build files - Makefile and the source .ht and .tcl files used to build the graphviz.org website are located in the graphviz-web subdirectory of the **document root directory**

## 4 Website Installation

The following steps were taken to install the new graphviz.org website.

### 4.1 Apache Configuration

4.1.1 In order to enable development of the new website, a test Apache server is configured to use port 8080. The Apache configuration file was modified to be different from the main Apache configuration file located at `/etc/httpd/conf/httpd.conf` as follows:

- The `ServerRoot` parameter commented out.
- The `Listen` value set to 8080 (binds to specific ip address and/or port)
- `User` and `Group` parameters commented out to prevent the server from being run as the apache user and group
- `DocumentRoot` parameter set to `/home/dperry/myhttpd/html`
- The `<Directory>` attribute set to `/home/dperry/myhttpd/html`

4.1.2 The test web server is started or stopped with the following commands:

```
/usr/sbin/httpd -d /home/dperry/myhttpd -k start  
/usr/sbin/httpd -d /home/dperry/myhttpd -k stop
```

4.1.3 The test web server is configured to use PHP by making the following changes to `httpd.conf`:

Verify the PHP module is loaded

```
LoadModule php5_module modules/libphp5.so
```

We want to allow `.php`, `.php2`, `.php3`, `.php4`, `.php5`, `.php6` and `.phtml` files to be executed as PHP, but nothing else

```
<FilesMatch "\.ph(p[2-6]?|tml)$">  
SetHandler application/x-httpd-php  
</FilesMatch>
```

4.1.4 Configure the test web server to enable Drupal url rewriting and clean urls by making the following changes to `httpd.conf`:

- Verify the rewrite module is loaded  

```
LoadModule rewrite_module modules/mod_rewrite.so
```

- Change the line  
Options Indexes FollowSymLinks  
to Options Include FollowSymLinks
- Change the line  
AllowOverride None  
to  
AllowOverrides All

## 4.2 MySQL Database Preparation

The following steps were taken to prepare a database for use by Drupal.

- Create a password for root user for the first time  
mysqladmin -u root password graphviz2011
- Create a database called graphviz  
mysqladmin -u root -pgraphviz2011 create graphviz
- Create stored procedures needed during the make process. Run the following two commands from directory /home/dperry/myhttpd/html/graphviz-web:  
mysql -u root -pgraphviz2011 graphviz < create\_select\_html\_sp.sql  
mysql -u root -pgraphviz2011 graphviz < create\_update\_html\_sp.sql

## 4.3 Drupal CMS Installation

- Drupal 6.19 was downloaded from drupal.org as a .tar.gz file and extracted into the **DocumentRoot** directory, /home/dperry/myhttpd/html.
- \$DocumentRoot/sites/default/default.settings.php was copied to settings.php in the same directory.
- The permissions on \$DocumentRoot/sites/default and settings.php were changed to writable.
- Start the web server and browse to index.php. This will invoke the Drupal installer. Follow the instructions and use the MySQL root user id and password from the MySQL Database Preparation section above when requested.
- Set the site name and email address from which the site will send out mail. Set the username, email, and password of the primary Drupal administrator account.
- An email will be sent to the administrator email address that will contain a link for a one-time login to the web server.

## 4.4 Mollom Spam Filter Setup

Once the website is functional, a mollom.com account needs to be setup in order to protect the website from spam.

- Download a Drupal plugin from mollom.com and extract it to the `$RootDirectory/sites/all/modules` directory.
- Setup a mollom free account with user id graphviz and password `!@mollom#$` The name on the account is Graphviz Software.
- Login to the mollom.com account
- Select "Manage sites" from the upper right menu
- Select "Add subscription" to create a new key pair.
- Visit the graphviz.org website's module list (*Administer >> Site building >> Modules*) and enable the Mollom module.
- Visit the graphviz.org website's Mollom settings page (*Administer >> Site configuration >> Mollom*) and enter the key pair from two steps above.
- Review the other Mollom settings and adjust as necessary.

## 4.5 Setup cron Job

Created a crontab file containing the following:

```
34 4 * * * curl --silent --compressed http://graphviz.org:8080/cron.php
```

## 5 Website Administration

### 5.1 User Setup

User roles are created to enable the assignment of web resource access permissions. The following roles are created:

- site administrator - granted permission to access all website resources
- graphviz-admin - granted permission to access all resources relevant to the graphviz website. Not granted permission to access resources associated with Drupal.
- anonymous user - granted permission to view web pages, submit a contact form, and to create a new account
- authenticated user - granted all permissions of anonymous user plus permission to create forum and wiki content
- graphviz-interest - granted all permissions of authenticated user plus permission to post comments without approval, upload file content, and create forum topics
- graphviz-team - granted all permissions of the graphviz-interest user plus permission to edit any forum topic, access freelinking list (used by wiki), create/edit web page content, and edit wiki content
- graphviz-dev - granted all permissions of the graphviz-interest user plus permission to create and edit blog entries, and access freelinking list

Table 1 sums up the permissions assigned to each role.

graphviz-admin, graphviz-dev, graphviz-interest, graphviz-team and site admin users are also authenticated users.  
 The terms nodes and page content refer to graphviz website pages  
 Revisions refers to previous versions of a web page

Permission	anonymous	authenticated	graphviz-admin	graphviz-dev	graphviz-interest	graphviz-team	site admin
access content	X	X					
administer content types							X
administer nodes			X				X
create page content			X			X	X
delete any page content			X				X
edit any page content			X			X	X
create wiki content		X					
delete any wiki content			X				X
delete own wiki content		X					
edit any wiki content			X			X	X
edit own wiki content		X					
administer forums			X				X
create forum topics			X	X	X	X	X
delete any forum topic			X				X
delete own forum topics				X	X	X	X
edit any forum topic			X			X	X
edit own forum topics				X	X	X	X
delete revisions			X				X
revert revisions			X				X
view revisions			X				X

Table 1

## 5.2 User Role Assignments

The following user ids are created and assigned to the listed roles. The initial password for each user is 'welcome'

graphviz-interest, graphviz-dev, graphviz-team

yifanhu

ellson

graphviz-interest, graphviz-dev, graphviz-team,  
 graphviz-admin, site administrator

north

erg

dperry

authenticated user

g\_auth

graphviz-team

g\_team

graphviz-dev

g\_dev

graphviz-interest  
g\_interst

### 5.3 Contact Form Assignments

The contact form allows users to submit comments or questions on a fixed set of categories. The categories and the email addresses used to forward the form are listed below:

Category	Recipients
general comment	graphviz-interest@research.att.com
graphviz features	graphviz-interest@research.att.com
graphviz interest group	graphviz-interest@research.att.com
contributing example graphs	yifanhu@research.att.com
build problems	ellson@graphviz.org
installation problems	ellson@graphviz.org
software configuration problems	ellson@graphviz.org
Windows related issues	arif@research.att.com
graph layout issues	erg@graphviz.org
website issues	dperry@research.att.com
circo, dot, fdp, neato, sfdp,twopi	graphviz-interest@research.att.com

Table 2

### 5.4 Administration

The Drupal database should be backed up on a regular basis. The command to backup the database is :

```
mysqldump -u root -pgraphviz2011 --opt graphviz > /home/dperry/db_backup.sql
```

If the database ever needs to be restored, the following command is used:

```
mysql -u root -pgraphviz2011 --database=graphviz < db_backup.sql
```

This backup command is scheduled to run with the following crontab entry:

```
40 4 * * * mysqldump -u root -pgraphviz2011 --opt graphviz > /home/dperry/db_backup.sql
```

## 6 *User Operations*

The operations that are permitted by a user are determined by the user's assigned roles. Table 1 lists valid operations assigned to each defined user role. This section describes details about permitted user operations.

### 6.1 *Anonymous User Operations*

Anonymous users can access web pages, view forum and wiki posts, submit a contact form and create a new account. When the **create new account** button is clicked, the **User account** page is displayed. The user then enters a username and e-mail address. Word verification is required before the form can be submitted. After the form is submitted, the system checks that the e-mail address is not currently in use. A confirmation e-mail is sent to the user that includes a system generated password and a link that can be used to log onto the website one time. After logging in, the user is directed to a page where the password can be changed.

The **Contact** form enables all users to submit a message to the website. It requires a name, e-mail address, subject, category and the message body. Word verification is required before the form can be submitted. The system sends the message to the e-mail address identified in Table 2 based on the category selected by the user. The category is selected from a pre-configured set.

### 6.2 *Authenticated User Operations*

Users who create a new account and who respond to the system generated e-mail, become authenticated. Authenticated users are permitted to create and update wiki content, view forum content and create comments.

#### 6.2.1 Wiki Support

Authenticated users can create and edit their own wiki content. When the user selects Create content > Wiki Page from the navigation menu, the Create Wiki Page form is displayed. The wiki form accepts a title, a set of Graphviz terms, a category, a list of tags, the wiki content and a file attachment. It is required to provide a title and at least ten characters within the message body. All other items are optional. The file attachment size is limited to 1 MB.

#### 6.2.2 Forum Support

Authenticated users are permitted to view content from any forum, but they cannot create content. They are permitted to add comments to any forum web page.

### 6.3 *Graphviz-interest User Operations*

Graphviz-interest users are authenticated users who have accepted an invitation to join the Graphviz interest group. They retain all permissions of authenticated users, and in addition

they are permitted to create forum content and to edit/delete their own forum content.

The graphviz-interest user creates a forum topic by selecting Create content > Forum topic from the navigation menu. The required fields are subject, forum name, topic content (Body) and word verifier. Optional fields include Graphviz terms, tags and file attachment.

#### *6.4 Graphviz-dev User Operations*

Currently, graphviz-dev users have the same permissions as graphviz-interest users. The role exists to accommodate external users authorized to perform privileged activities not allowed by the graphviz-interest role.

#### *6.5 Graphviz-team User Operations*

The graphviz-team group consists of internal Graphviz developers and support personnel. Members of this group have all permissions assigned to graphviz-interest group members. They are also permitted to

- Create and edit any website content
- Edit any wiki content
- Edit any forum topic

#### *6.6 Graphviz-admin User Operations*

Graphviz-admin users have all of the permissions of graphviz-team members. They are also permitted to

- Administer web pages
- Delete web pages
- Delete wiki pages
- Administer forums
- Delete forum topics
- Delete/Revert/View web page revisions

#### *6.7 Site Administrator User Operations*

The site administrator has all the permissions assigned to the graphviz-admin role. They can also administer web page content types and perform many tasks that pertain to website configuration and maintenance outside of a Graphviz context.

## 7 Building the Graphviz Website

The Graphviz build script has been changed slightly. A few rules have been modified and a new rule has been added.

### Rule

.ht.php:

```
./ht2php.py "${PAGESET}" $<
```

```
./update_html.sh $@ | mysql -u root -pgraphviz2011 -D graphviz
```

### Rule Change:

The second line of this rule is new. It invokes a stored procedure on the MySQL database that stores the content of the associated .php file. Since the website is database driven it is necessary to transfer the web pages from the DocumentRoot subdirectory to the database. The .php files are created in exactly the same way that they are now. The new step only inserts them into the database.

### Rule

.PHONY: Download\_source.ht Download\_linux\_fedora.ht Download\_linux\_rhel.ht  
Download\_linux\_ubuntu.ht Download\_solaris.ht Download\_macos.ht Download\_windows.ht  
Download\_att.ht pdf doc **Download..php**

### Rule Change:

Since the Download..php file is not a member of \${PAGESET}, it must be inserted into the database by a new rule. This rule must be executed even though it does not affect the target.

### Rule

Download..php:

```
./update_html.sh Download..php | mysql -u root -pgraphviz2011 -D graphviz
```

### Rule Change:

This new rule inserts the Download..php file into the database.

### Rule

get\_source:

```
./select_html.sh ${PAGESET}
```

### Rule Change:

This is a new rule that accomplishes the following tasks:

- Extracts the web pages from the database
- Removes content generated by the make process that is common to all pages

- Removes content that is dynamically generated during the make process
- Inserts the page specific content into the .ht and .tcl files used by the make process

The purpose of this rule is to capture changes made to the web pages within the Drupal page editor ( either the plain text editor or the WYSIWYG editor ). After running make with this rule, the standard make process can be run to refresh the website with changes made to the common page content, the dynamic page content or changes to the content extracted from the database. If it is desired to make changes to the web pages on the server, this rule should be run beforehand in order to capture and preserve any changes that were made to the database. Otherwise changes made from within Drupal will be overwritten. Changes to common content and dynamically generated content can be made before this rule is run. Note that changes made on the server to page specific content will be overwritten with content stored on the database. It is therefore important to make page specific edits on the server after this rule is run.

## **8 Webpage Editing with Drupal WYSIWYG**

Users assigned to the graphviz-team , graphviz-admin and site administrator roles are permitted to create and edit web pages. Editing functions are accessed by selecting the “Edit” tab located at the top of the web page.

The edit view has several sections including

- Vocabularies
- Menu Settings
- Body
- Input Format
- Meta Tags
- Revision Information
- URL Path Settings
- Comment Settings
- File Attachments
- Authoring Information
- Publishing Options

The body section will be the section of interest when the content of the web page needs to be modified. The body section contains a WYSIWYG editor. There are tools for basic web page editing located at the top of the body section. This WYSIWYG editor can be switched to a plain text editor by clicking the *Switch to plain text editor* link below the Body section.

There are some restrictions on edits made with the Drupal editors. Graphviz web pages are created by a make process. They contain template content common to all pages, dynamically generated html and content that is page specific. It is necessary to delimit the

page specific content in order to extract any modifications to it from the Drupal database.

Pages not generated from .tcl files have this content delimited by <!-- body --> and <!-- trailer -->. It is required that any changes made with the Drupal editor occur between these two tags. Otherwise the changes will be removed during the make process.

Pages that are generated from .tcl files include all pages with navigation links in the Download sub-menu:

- source
- fedora
- macos
- rhel
- solaris
- ubuntu
- windows

When editing these web pages with the Drupal editor, place changes between the tags <!-- cut1 --> and <!-- cut2 -->. These two tags must be on separate lines in order for the sed command to use them properly.

After making changes with the Drupal WYSIWYG editor, click on the *Source* button located in the upper left corner of the toolbox or use the *Switch to plain text editor* located in the lower left corner of the Body section. The editor display will change to a text editor and it will be possible to verify that all changes occur within the delimiting tags. If changes appear outside of the delimiting tags, just cut and paste them between the tags.

After page changes are made and it has been verified that they are placed between the delimiting tags, click on the *SAVE* button to update the Drupal database. If you want to save the previous version of the page, be sure to check the ***create new revision*** option under the ***Revision Information*** section.

## 9 Saving Changes to CVS

Web page editing is currently done with a plain text editor on .ht and .tcl files checked out from the cvs repository. It is still possible to edit content in this way for the new website as well. These files contain the web page content that is specific to each page. During the make process, tcl scripts are run to wrap dynamic graphviz version information around the specific content. The output of these tcl scripts is written to files with an .ht extension. The content of all files with a .ht extension is then wrapped with html code common to all pages and output to like named files with a .php extension. These .php files are then made accessible to the web server.

The new Drupal based website accesses web pages from a MySQL database. The pages can be modified with Drupal editors and then exported to the same flat .tcl and .ht files used currently. The make process has been changed so that the generated .php files are imported into the Drupal database. Therefore changes made by a text editor to the .ht or .tcl files appear on the website exactly as before.

A new rule has been added to Makefile that exports the web pages from the Drupal database, strips out the specific content of each page and then inserts it into the appropriate place in the corresponding .ht or .tcl file. This enables changes to web pages made within a Drupal editor to be checked into cvs. The command to export the current web pages from Drupal is ***make get\_source***. It will update the .ht and .tcl files with the corresponding content from the Drupal database. In order to synchronize cvs with the Drupal database, check-in the Graphviz web page source files after running make get\_source.