

Mission: Impossible

Content Management

Your Mission:

Your mission, should you choose to accept it: deploy a enterprise-class content management solution that runs entirely on your existing web server environment, supports multiple sites, maintains existing user access, security, and disk quota protections, is easy for non-technical contributors to learn and use while being flexible enough to handle a wide range of sites, and is relatively simple for IT to maintain--while still being expected to do your day-to-day job functions, as well. All of this, of course, is expected on a budget of zero.

Staring

Jason Pitoniak

Technical Team Lead

Enterprise Web Applications

Information & Technology Services

Rochester Institute of Technology

jason.pitoniak@rit.edu @kodiak324

The Web at RIT

- No central marketing department
- University Publications oversees homepage and second-level content
- Individual departments responsible for their own content
- No “web” budget lines

Development Options

- Build internally
- Hire a student
- Hire on-campus resources
- Hire an off-campus developer

Build Internally

- Usually means buying a staff assistant a copy of Dreamweaver and, if she's lucky, a \$20 pre-made HTML template
- Some colleges have dedicated web staff
- Hire a web developer or hire new faculty?

Hire a Student

- We have very talented students with lots of great ideas
- Hiring a co-op for a quarter to design and build a site is a great deal financially
- Students don't stay forever
- What happens to the site when they leave?

Use On-Campus Resources

- We have a group on campus that provides web development services on a chargeback basis
- Prices on par with co-op salary
- They understand our web standards
- High demand, often backlogged

Hire an Off-Campus Developer

- Departments use everything from freelance developers to big-name agencies
- Generally the least cost-effective option

Leaving us with sites that run the
gamut from this...

GLOBAL VILLAGE

[About](#) | [Housing](#) | [Learning](#) | [Deals](#) | [Contact](#) | [Finance & Administration](#)

[DINING](#)
[SHOPPING](#)
[SERVICES](#)
[EVENTS](#)

WELCOME!

Global Village is an internationally inspired, LEED-certified sustainable residential community and commercial complex located on RIT's campus. Open to both the public and RIT community, RIT's Global Village combines state-of-the-art student living with dining, shopping, learning, and event spaces to create a unified and modern upscale community. Global Village also hosts a number of convenient services including a post office, print center, and salon, as well as Shop One®, RIT's acclaimed fine art and craft gallery.

[Explore Global Village today!](#)



LIVE & STUDY



DINE & SHOP



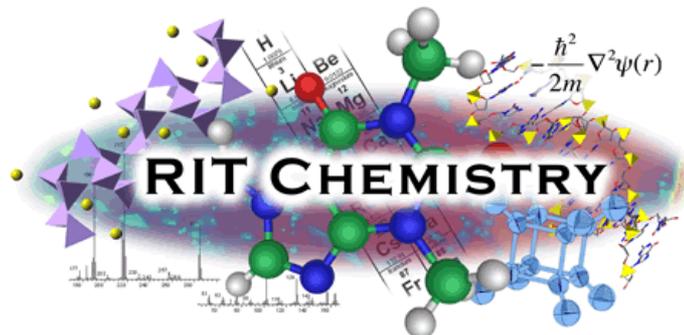
WORK & PLAY



UPCOMING EVENTS

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

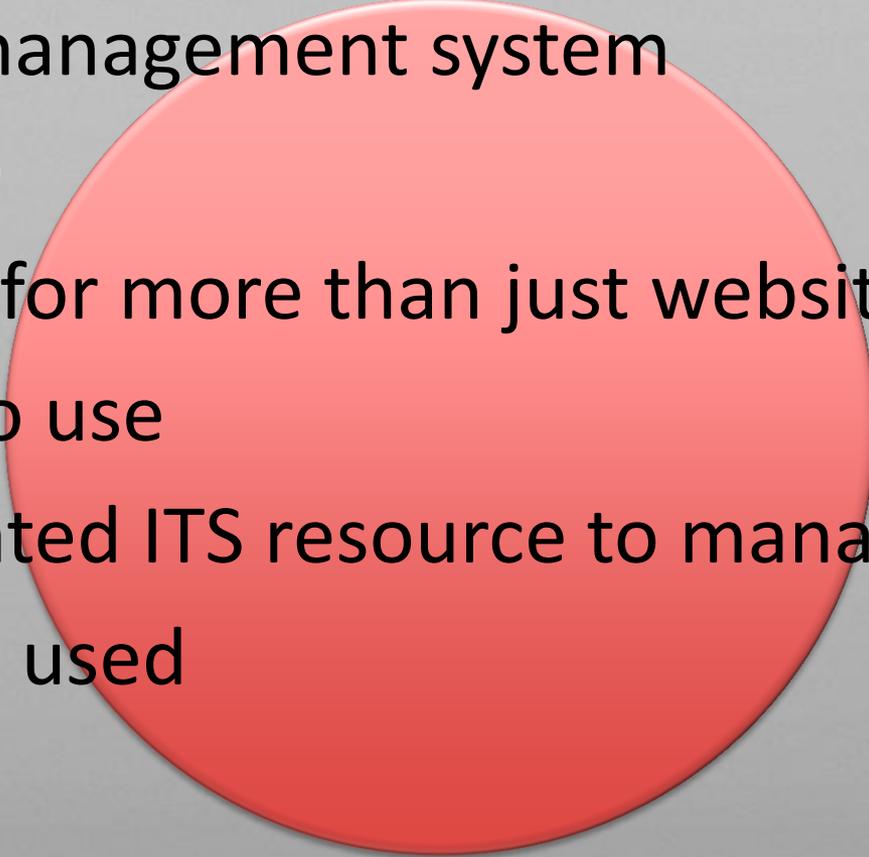
...to this...

[Department Info](#)[Faculty & Staff](#)[Research](#)[Co-op](#)[Courses and Programs](#)[Links](#)

Department of Chemistry
85 Lomb Memorial Drive
Rochester Institute of Technology
Rochester, NY 14623

voice: 585.475.2497
fax: 585.475.7800
email: aemsch@rit.edu

Content Management at RIT

- Early-2000's we purchased a commercial content management system
 - Expensive
 - Designed for more than just websites
 - Difficult to use
 - No dedicated ITS resource to manage
 - Didn't get used
- 

New Web Environment

- 2007-2008 we replaced our antiquated web server with a new web environment
- Added new features that were expected by our customers
 - PHP 5 (old server was still running PHP 3!)
 - MySQL databases
 - Clean URLs instead of department-number based account names
(i.e. <http://www.rit.edu/~491www>)
- Introduced staging and production environments

Moving Forward

- 2009 University Publications published a new set of web design standards
- The Web Advisory Committee decided it was time to revisit CMS

CMS Selection

- We looked at what was already being used on the web environment
- Surveyed web stakeholders about features they'd like to see in a CMS

No Budget

- Selected solution had to be free
- Selected solution had to run in the current environment
- Selected solution had to be easy to get up and running with minimal training
- But we also had to consider the commercial package we were still licensing

The Contenders

- Joomla
- Drupal
- The commercial CMS named for the colored shapes used in its UI that we are still not disclosing

Joomla

- Used by Undergraduate Admissions
- Open Source
- Flexible
- Good extension availability
- PHP, MySQL
- No multisite support

Drupal

- Already very popular on the web environment
- Extremely extensible
- Huge developer community
- Slow, bloated, somewhat complex

“Crimson Circle” CMS

- Out-of-the-box support for multiple sites
- Much more than a simple website manager
- Difficult to use
- Requires Windows server
- Works better in IE than other browsers
- Expensive annual contract
- Required training to use effectively, but no training budget was available

R·I·T

And the Winner Is...



Why Drupal?

- Easy enough for non-technical users to learn
- Flexible enough to power our more complicated sites
- E-Commerce options available
- Students know it

Implementation Team

- ITS
 - Technical Lead
 - Senior Programmer/Analyst
 - Project Manager
- University Publications
 - Two Web Developers
- Core team was responsible for decision making, implementation, evangelism, and training

Additional Resources

We also pulled in additional resources from other areas with professional web development staffs, as needed:

- Planning, requirements gathering
- Training
- Pre-release piloting

The Challenges

Challenge 1

Technical Lead becomes FBI Agent

- Really. I couldn't make that up.
- Forced the programmer/analyst to take on a lot of additional responsibilities
- Forced the programmer/analyst to work a lot of weekends!

Challenge 2

Fitting Drupal into our existing URL structure

Existing Setup

- Each site exists in its own Unix account
- Mod_rewrite rewrite map files with internal redirects are used to map clean URLs to these accounts
- This setup allows us to maintain disk quotas, isolate processes (suexec), and easily disable sites with security or other issues when necessary

Drupal

Drupal would, ideally, be installed in one account for easier maintainability...but where?

Homepage Account

- We weren't planning on rolling out Drupal for the top-level content, at least not initially
- While mixing Drupal with static content would work, it would be confusing and difficult to manage
- Drupal would be maintained by ITS, homepage is maintained by UPubs

Add Sites to RewriteMap Files Manually

- Webman (our web management system) overwrites the map files when new sites are added
- Reworking Webman was not an option at the time

Create New RewriteMap Files for Drupal

- Worked, but all Drupal content was stored in a single account
- No quota control
- No easy way to create multiple MySQL databases
- Difficult to take down a single site

Continue Using Individual Site Accounts

- Developed an .htaccess file that routed requests to the central Drupal installation
- Allowed us to continue using existing controls
- Made retrofitting existing sites easier

.htaccess Example

```
shell01off01.rit.edu - PuTTY

### SETTINGS FOR DRUPAL - DO NOT EDIT ###
SetEnv REQUESTED_SITE /fa/globalvillage

RewriteEngine on
RewriteBase /fa/globalvillage
RewriteRule ^install.php /~w-drupal/install.php [L,QSA]
RewriteRule ^update.php /~w-drupal/update.php [L,QSA]
RewriteRule ^cron.php /~w-drupal/cron.php [L,QSA]
RewriteRule ^xmlrpc.php /~w-drupal/xmlrpc.php [L,QSA]
RewriteRule ^misc($|(/.*)) /~w-drupal/misc$1 [L,QSA]
RewriteRule ^themes($|(/.*)) /~w-drupal/themes$1 [L,QSA]
RewriteRule ^modules($|(/.*)) /~w-drupal/modules$1 [L,QSA]
RewriteRule ^sites/all($|(/.*)) /~w-drupal/sites/all$1 [L,QSA]
RewriteRule ^sites/rit.edu.fa.globalvillage($|(/.*)) /~w-drupal/sites/rit.edu.fa.globalvillage$1 [L,QSA]
RewriteRule home/w-gv/DrupalFiles($|(/.*)) system/files$1 [L,QSA]
RewriteRule ^user/login($|(/.*)) /~w-drupal/rit-sso.php [L,QSA]
RewriteRule ^ritlogin($|(/.*)) /~w-drupal/rit-sso.php [L,QSA]

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^.*$ /~w-drupal/index.php?q=$1 [L,QSA]

RewriteCond %{REQUEST_URI} ^/~w-gv/?$
RewriteCond /home/w-gv/www/index.html? !-f
RewriteCond /home/w-gv/www/index.php !-f
RewriteRule ^$ /~w-drupal/index.php [L,QSA]
### END - SETTINGS FOR DRUPAL ###
shell01off01{jrpisd}106: █
```

Drupal Side

- PHP sees the URL as coming from the redirected URL (i.e. /~w-drupal/index.php), so we set a REQUESTED_SITE environment variable to determine which site was requested
- Set an auto_prepend_file in w-drupal's php.ini to modify environment variables based on REQUESTED_SITE

Prepend.php

shell01off01.rit.edu - PuTTY

shell01off01{jrpisd}106: more /home/w-drupal/php_data/prepend.php

```
<?php
$rs = '';
foreach($_SERVER as $k => $v) {
    if(substr($k, -14) == 'REQUESTED_SITE') {
        $rs = $v;
        break;
    }
}

if(!empty($rs)) {
    //rewrite server variables to reflect original path
    $_SERVER['SCRIPT_NAME'] = str_replace('/~w-drupal', $rs, $_SERVER['SCRIPT_NAME']);
    $_SERVER['PHP_SELF'] = str_replace('/~w-drupal', $rs, $_SERVER['PHP_SELF']);

    //adjust session cookie handling
    ini_set('session.cookie_path', $rs);
    ini_set('session.name', 'PHPSESSID_' . str_replace('/', '_', $rs));

    unset($rs);
}
else {
    //default site
    $_SERVER['SCRIPT_NAME'] = str_replace('/~w-drupal', '/its/drupal', $_SERVER['SCRIPT_NAME']);
    $_SERVER['PHP_SELF'] = str_replace('/~w-drupal', '/its/drupal', $_SERVER['PHP_SELF']);
}

function getConfigValue( $configName ){
--More-- (74%)
```

Challenge 3

Protecting Users from Themselves

Access Levels

- We wanted to give site administrators as much control over their site's as possible
- As the “Innovation University” we want to allow and encourage people to be innovative with their sites
- History tells us that being too restrictive leads people to buy their own “servers” to run in their offices
- However...

Access Controls

- ...we knew that if we gave out too much control, some users would change things they shouldn't (intentionally or unintentionally)
 - Undo security settings
 - Change file paths to locations with insufficient privs
 - “I don't want ITS messing with my site!”

The Solution

- In talking with a developer at NTID, we found that they had created a custom module that blocked users from modifying or deleting “user 1”
- We expanded their code into our rit_admin module:
 - Hides form fields for settings that users should not change
 - Blocks users from turning off essential modules

Challenge 4

File System Woes Part I: Private Files

File Access

- By default, Drupal stores files in a “files” directory in a given site’s configuration directory (i.e. sites/rit.edu.fa.globalvillage/files)
- Since we needed to store files in a different account, it made sense to use symlinks

```
shell01off01{jrpisd}108: cd /home/w-drupal/www/sites/rit.edu.fa.globalvillage/  
shell01off01{jrpisd}109: ls -la  
total 26  
dr-xr-xr-x+  3 w-drupal w-drupal-group    1024 Sep 13 23:30 .  
drwxrwsr-x+ 65 w-drupal w-drupal-group    3072 Sep 12 14:45 ..  
lrwxrwxr-x   1 w-drupal w-drupal-group     22 May 18 11:34 files -> /home/w-gv/DrupalFiles
```

Backup Issues

- While we were testing the symbolic link idea, we started seeing issues with our backup system
- Symlinks were blamed, leading us to explore alternative methods

Public and Private Files

- Drupal has a concept of “public” and “private” files
- Public files are accessible by the web server and can be accessed without getting Drupal involved
- Private files are not directly web accessible, Drupal opens the file and sends its contents to the browser

Switch to Public Files

- We changed the Drupal sites to use private files with an absolute path to the DrupalFiles directory and all was good.
- Or so we thought...

Problem 5

Issues with Cache Headers

Show Me the Cache

- With private files enabled, images were not being cached by browsers
- Turns out cache headers were being sent, but they were already expired.

Brainstorm!

- We realized that, in our .htaccess configuration, requests only forward to Drupal they don't map to an actual file
- So if there was really a file located within system/files then Apache, not Drupal, would handle it
- We switched sites to physically store files in a system/files directory structure and everything started working as we wanted...

...almost

Challenge 6

Developers need access to themes

Themes

- We found an acceptable solution for files, but we also wanted to give developers access to their themes directories
- Unlike “files,” the path to the themes directory cannot be changed
- Not being able to use symlinks posed a real problem

Symlink Problems? What Symlink Problems?

- We talked to our systems administrators to see what it would take to get our backups to work with symlinks
- Did some testing and discovered that symlinks weren't really a problem.
- We started testing and symlinks worked well!

Challenge 7

Symlinks don't push to production

Webman Issues

- In our environment, site admins don't have direct access to production. They must use the transfer app in Webman to move files to production.

Setup Script

- Wrote a quick PHP script that collected site info in a form and created the directory structure for the site from a template.
- But that was just one small step in a complicated installation process...

Challenge 8

Setup Time Concerns

Multi-Step Process

- It took about 15 steps to install Drupal:
 - Create website account, if necessary
 - Create MySQL database
 - Copy MySQL passwords to Drupal account
 - Set up settings.php and symlinks (now automated)
 - Set up .htaccess (search and replace through a template)
 - Run install.php or copy database
 - Add users and site-specific settings
- This process would take 30-45 minutes for an experienced installer to complete
- Error Prone

There Must Be A Better Way

- We needed better automation
 - Reduce time to install
 - Reduce error rate
 - Eventually turn setup process over to EWA student employees and/or service desk staff

Drupalizer

- With a few hours of scripting “drupalizer” was born
 - Version 0.1: simple script that pulled site info from environment variables and created an .htaccess
 - Version 1.0: Added a Web UI to specify options, created directories and .htaccess
 - Version 1.1: “Wizard” interface posts to a second set of scripts to do back end configurations

Streamlined Process

Site setup is now down to six steps:

- Create account
- Upload Drupalizer
- Run Drupalizer
- Copy database or run install.php
- Set site specific settings
- Add users

Drupalizer Demo

Current State

- After just over a year and half since launch, we've had about 65 sites convert to Drupal
 - Up from about 20 sites a year ago
 - Roughly 10% of all the sites hosted on the web environment are now in Drupal
- A similar setup for NTID hosts an additional 32 sites

Current State (Continued)

- One college has converted all of their sites to Drupal and at least two others are in the process
- At least one site in each college is using Drupal, as are the President's and Provost's offices
- Plans are underway to convert 30+ Finance and Administration divisional sites to Drupal

Next Steps

- Upgrade to Drupal 7
- Build a new environment for running Drupal
 - Eliminate single points of failure (MySQL, NFS)
 - Reverse Proxy Cache for improved performance and scalability
- Drupalizer Improvements
 - Auto-install database from template
 - Set site specific settings during setup
- Drupal “sudo”
- Better search capabilities, possibly using a new enterprise search tool

Thank You

Jason Pitoniak

jason.pitoniak@rit.edu

@kodiak324

Red staplers are good!

TPR9



“Impossible” font available from dafont.com.