

Splitting and Merging

Down and Dirty Made Easy

Splitting WebGUI Sites

E unus pluribus

Splitting WebGUI Sites

- [Two possible ways

- Duplicate and cut-down

- No need for messy rewrite rules

- Entirely separate site to spread over server load

- Easier to accomplish

Splitting WebGUI Sites

- [Two possible ways

- Make a "Virtual" site

- Using Apache's `mod_rewrite`

- No duplication of assets or files

- Simplifies tricks between sites like SSO

Duplicate + Cut-Down

A site divided against itself

Duplicate and Cut-Down

— [Far easier than any messy database / file operations to physically split a WebGUI site

— [Not as flexible as Virtual sites

— [Do as much inside WebGUI as possible

— [Keep only the assets and files we need

— [Complete copy of everything else

Set the Stage

— [www.plainblack.com

— [Change www.plainblack.com/software/webgui into
www.webgui.org

Setting Up

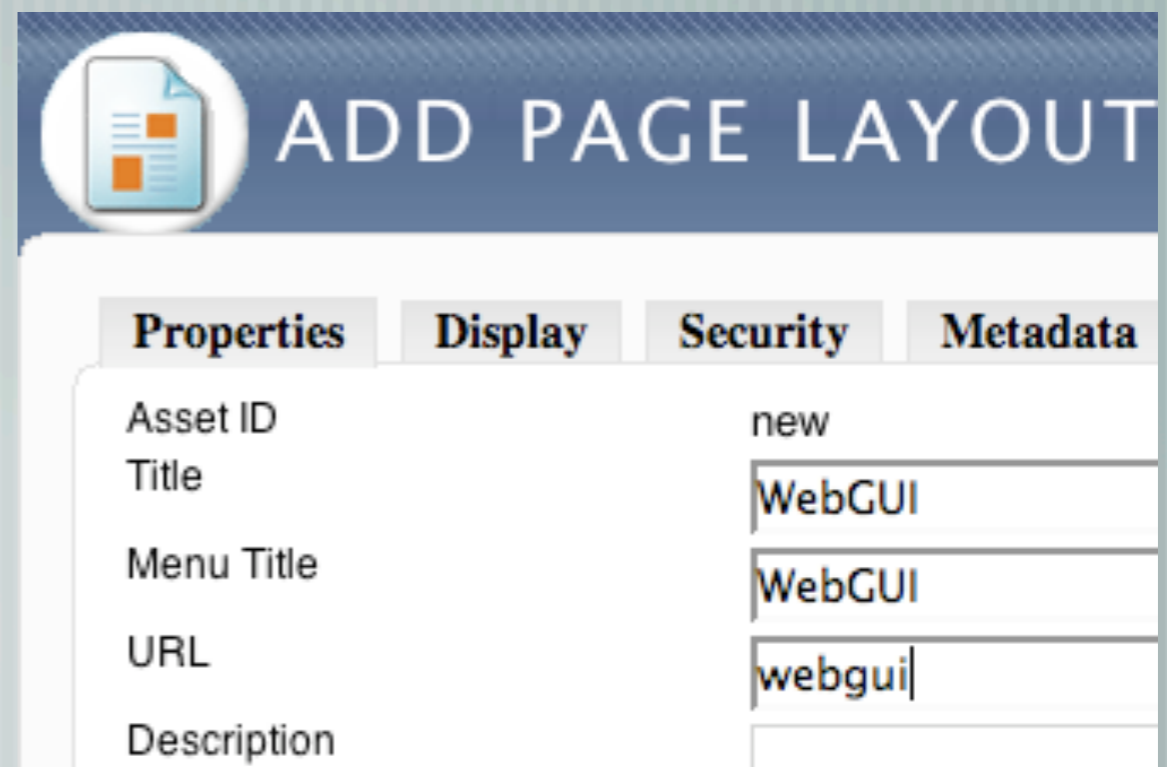
Make a new Page Layout to hold our entire site

Cut and paste our new site's content into this Layout



The screenshot shows the 'ASSETS' management interface. At the top, there is a header with a server icon and the word 'ASSETS'. Below the header is a table with columns for 'Rank' and 'Title'. The table lists three assets with ranks 1, 2, and 3. Each asset has a checkbox, a rank number, and a 'Title' field. The first asset has a rank of 1 and a title of 'new'. The second asset has a rank of 2 and a title of 'WebGUI'. The third asset has a rank of 3 and a title of 'webgui'. Each asset also has a 'Title' field and a 'Description' field.

Rank	Title
1	new
2	WebGUI
3	webgui



The screenshot shows the 'ADD PAGE LAYOUT' interface. At the top, there is a header with a document icon and the text 'ADD PAGE LAYOUT'. Below the header is a form with tabs for 'Properties', 'Display', 'Security', and 'Metadata'. The 'Properties' tab is selected, and the form contains fields for 'Asset ID', 'Title', 'Menu Title', 'URL', and 'Description'. The 'Asset ID' field is set to 'new', the 'Title' field is set to 'WebGUI', the 'Menu Title' field is set to 'WebGUI', and the 'URL' field is set to 'webgui'.

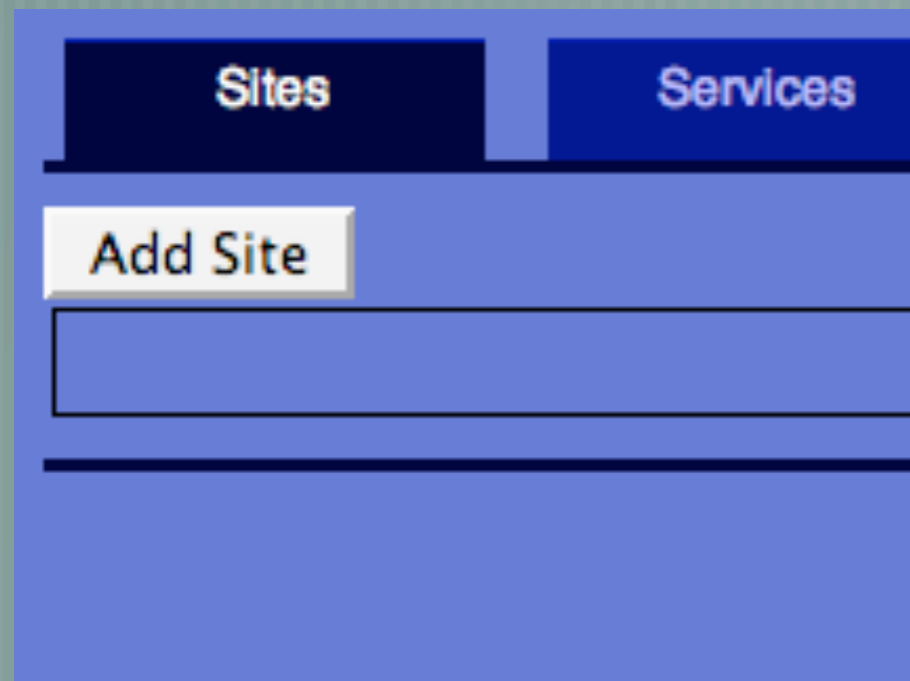
Asset ID	Title	Menu Title	URL	Description
new	WebGUI	WebGUI	webgui	

Duplicate and Cut-Down

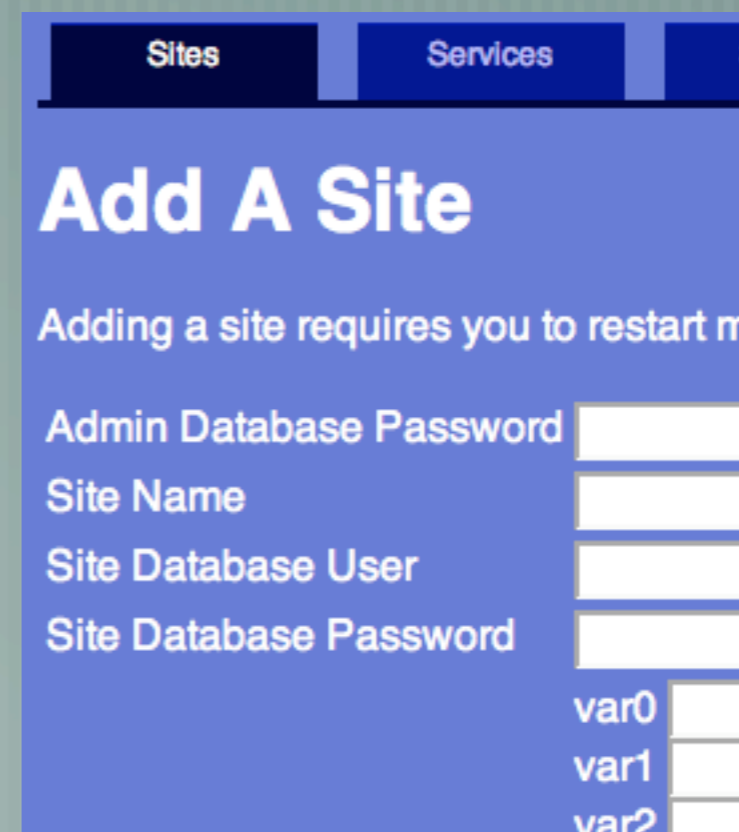
— [Create a new site to copy over

```
addsite.pl --adminPassword=<> --sitename=www.webgui.org
```

wreconsole.pl



The screenshot shows a web interface with a blue background. At the top, there are two tabs: 'Sites' (selected) and 'Services'. Below the tabs, there is a button labeled 'Add Site'. Below the button, there are three empty input fields for form data.



The screenshot shows a web interface with a blue background. At the top, there are three tabs: 'Sites' (selected), 'Services', and 'S'. Below the tabs, the title 'Add A Site' is displayed. Underneath the title, there is a warning message: 'Adding a site requires you to restart m'. Below the warning, there are four input fields: 'Admin Database Password', 'Site Name', 'Site Database User', and 'Site Database Password'. At the bottom right, there are three additional input fields labeled 'var0', 'var1', and 'var2'.

Duplicate

Database

```
mysqldump -u<> -p<> www_plainblack_com | mysql -u<> -p<> www_webgui_org
```

Files

```
cp -r /data/domains/www.plainblack.com/* /data/domains/www.webgui.org
```

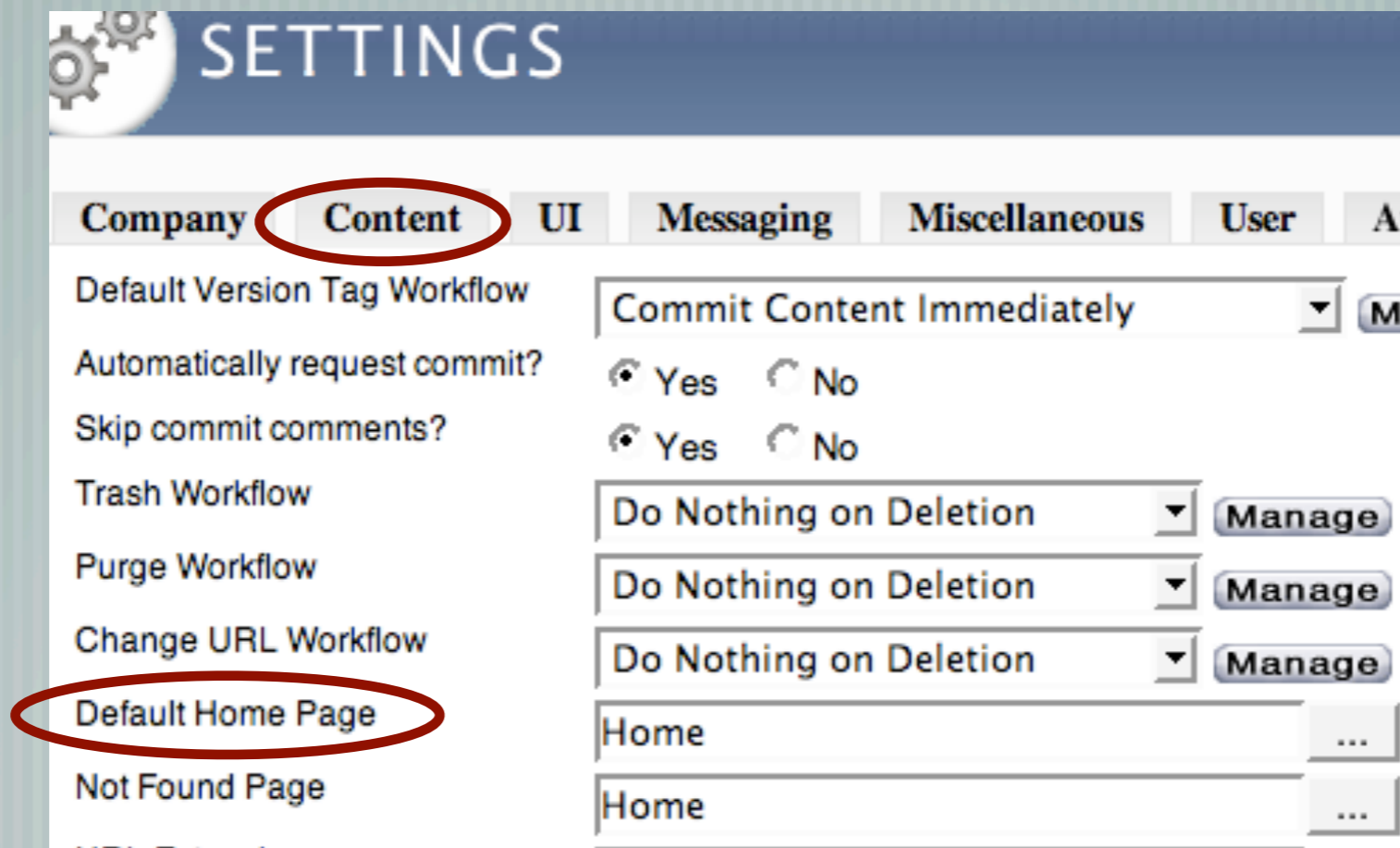
Clear the Cache whenever raw changes to the database are made

```
rm -rf /tmp/WebGUICache
```

Restart and test

and Cut-Down

- [Delete all the assets we don't need from both sites
- [Set our Home page for the new site



SETTINGS

Company **Content** UI Messaging Miscellaneous User Au

Default Version Tag Workflow Yes No

Skip commit comments? Yes No

Trash Workflow

Purge Workflow

Change URL Workflow

Default Home Page

Not Found Page

DONE!

Making a Virtual Site

A Little Slight of Regex

Making a Virtual Site

- [Two domains, One database / files

- [Apache mod_rewrite

- Makes setup invisible to end-user

- [Additional Benefits

- No duplication of shared assets / files

- SSO using new Operation ssoViaSessionId

Making a Virtual Site

- [Disadvantages

- More complicated to set up

- Shared statistics

Set the Stage

— [www.plainblack.com

— [Change www.plainblack.com/software/webgui into
www.webgui.org

Making a Virtual Site

First: Create a new, top-level Page Layout asset

Will act as our "Home" page for our new site

Will hold everything for www.webgui.org

URL: www.plainblack.com/webgui



<input type="checkbox"/>	Rank		Title
<input type="checkbox"/>	1	Edit More	+ Import Node
<input type="checkbox"/>	2	Edit More	+ Home
<input type="checkbox"/>	3	Edit More	Media



Properties	Display	Security	Metadata
Asset ID	new		
Title	WebGUI		
Menu Title	WebGUI		
URL	webgui		
Description			

Making a Virtual Site

Move all our content to our new Page Layout

Use existing "cut" and "paste" operations


All content from www.plainblack.com/software/webgui is now under /webgui

Root > Home

<input type="checkbox"/>	Rank		
<input type="checkbox"/>	3	Edit More	+ Gett
<input type="checkbox"/>	4	Edit More	+ Wha
<input type="checkbox"/>	5	Edit More	+ The
<input checked="" type="checkbox"/>	6	Edit More	+ Tell
<input type="checkbox"/>	7	Edit More	+ Site
<input type="checkbox"/>	8	Edit More	Welc
<input type="checkbox"/>	9	Edit More	Ad
<input type="checkbox"/>	10	Edit More	Key

Delete | Cut | Copy | Duplicate

Clipboard

- Select All
-  Tell A Friend

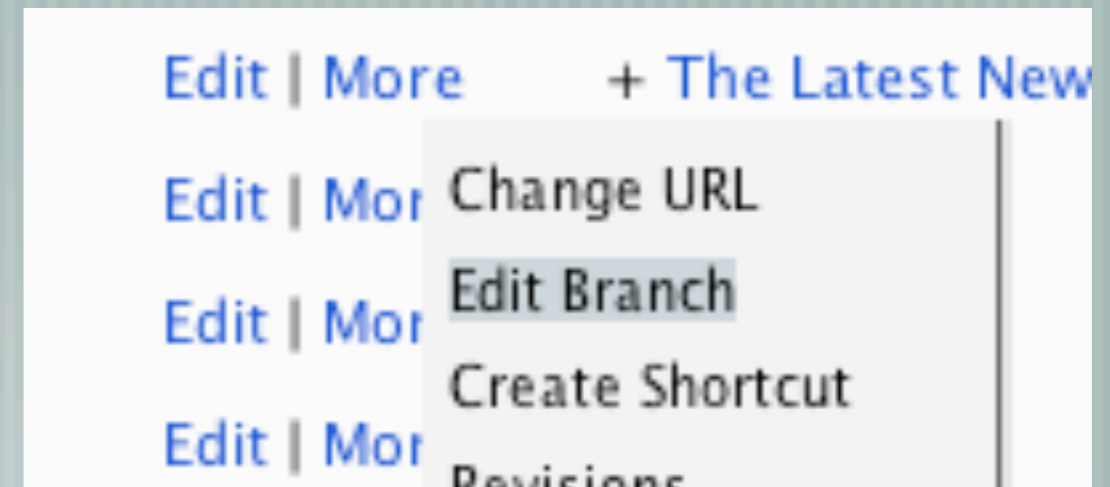
Paste

Normalize URLs

— [Ensure all our new site's asset's URLs begin with /webgui

— [Necessary for the mod_rewrite rules

— [Edit Branch



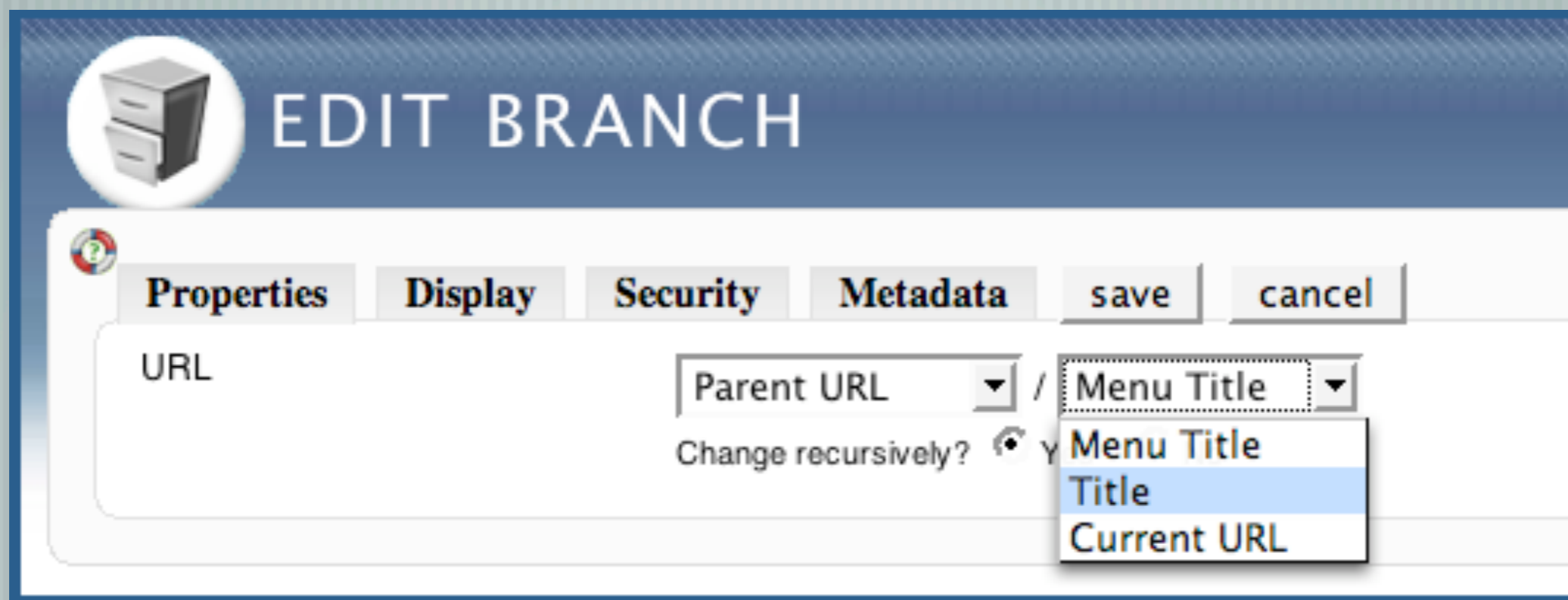
Normalize URLs

Parent URL is required

Menu Title or Title creates a new URL

Current URL will keep the old URL with the new domain

Change recursively: Yes!



Making a Virtual Site

- [Site cleanup

- Commit our changes

- Test our new site

- www.plainblack.com/webgui

- Change style template / other things with Edit Branch

- Prepare to write our `mod_rewrite` rules

Configure Apache

```
<VirtualHost *:80>
  ServerName www.plainblack.com
  ServerAlias plainblack.com
  DocumentRoot /data/domains/www.plainblack.com/public

  RewriteEngine On
  # block IIS related attacks from getting to WebGUI
  RewriteRule ^/scripts/ - [L]
  RewriteRule ^/default.ida - [L]
  RewriteRule ^/_vti_bin - [L]
  RewriteRule ^/_mem_bin - [L]

  # proxy webgui pages
  RewriteRule ^/(.*) http://%{HTTP_HOST}:8081/$1 [P]
</VirtualHost>
```

Configure Apache

Add domain to ServerAlias

```
<VirtualHost *:80>  
  ServerName www.plainblack.com  
  ServerAlias plainblack.com webgui.org www.webgui.org  
  DocumentRoot /data/domains/www.plainblack.com/public
```

Whitespace-separated list of domain names

Only necessary for modproxy under WRE

Now Apache knows it's serving for two, but both point to the same place

Configure Apache

- [RewriteEngine On

```
RewriteEngine On
```

- [Inside the `<VirtualHost *:80>` block

- [Turns on `mod_rewrite`

- [Only needed once per `VirtualHost`

Configure Apache

— [RewriteRule <match> <replace> [options]

— Exchange any <match> in the URL with <replace>

— Only matches against the path section of the full URL

— [RewriteCond <value> <match> [options]

— Only perform the next RewriteRule if the RewriteCond passes

— Many special variables to match against

Add Rewrite Rules

Add these rules after other RewriteRules and Redirects

```
# Handle webgui.org URLs
RewriteCond ${HTTP_HOST} ^(www.)?webgui.org$
RewriteRule ^/(.*)$ /webgui/$1

# proxy webgui pages
RewriteRule ^/(.*) http://%{HTTP_HOST}:8081/$1 [P]
```

It may take some tweaking to get the order of your RewriteRules correct

Use RewriteLog and RewriteLogLevel to debug

Configure Apache

Restart and Test

May have to add our new hostname to `/etc/hosts`

```
127.0.0.1 localhost plainblack.com www.plainblack.com webgui.org www.webgui.org
```

`http://www.webgui.org` should now bring us to our new site

Configure Apache

— [... but www.plainblack.com/webgui shows our new site too!

— We can redirect all calls to the new site

```
RewriteCond ${HTTP_HOST} ^(www.)?plainblack.com$  
RewriteRule ^/webgui/(.*)$ http://www.webgui.org/$1 [RL]
```

— [... but bookmarks to the old URL will all be broken!

— We can redirect the old URL to the new URL

```
RewriteCond ${HTTP_HOST} ^(www.)?plainblack.com$  
RewriteRule ^/software/webgui http://www.webgui.org/$1 [RL]
```

Merging WebGUI Sites

E pluribus unum

Merging WebGUI Sites

— [Easier to split than to merge

— [Thank GUIDs

— 1 in 64^{22} or $5.4 * 10^{39}$ or 54,000,000,000,
000,000,000,000,000,000,000,000,000,000

— Better chance of hitting the multi-millions lottery trillions
upon trillions of times

— Abstractions don't do it justice

Set up

— [Two sites: www.plainblack.com and www.webgui.org

— [Merge to form www.plaingui.com

— [Create a blank site to hold our merger

— Can move the merged site to replace one of our old sites later

— All we need to do to start over is delete the merged site

```
addsite.pl -adminPassword=<> -sitename=www.plaingui.com
```

Make a base

— [Start with a base of one of the sites

— Database

```
mysqldump -u<> -p<> www_plainblack_com | mysql -u<> -p<> www_plaingui_com
```

— Files

```
cp -r /data/domains/www_plainblack.com /data/domains/www_plaingui.com
```

Prepare to Merge

- [Prepare the other site to be merged in

- [Users and Groups

- Groups with the same groupId should be merged

- Users with the same username should have the same userId

Fix User IDs

```
#!/data/wre/prereqs/perl/bin/perl
use strict; use DBI;

my $main = DBI->connect("DBI:mysql:dbname=www_plaingui_com","root","password",{RaiseError=>1});
my $merge = DBI->connect("DBI:mysql:dbname=www_webgui_org","root","password",{RaiseError=>1});

# Get all the users from the main db
my $users = $main->selectall_hashref("select username,userId from users","username");
for my $username (keys %{$users}) {
    my $newUserId = $users->{$username}->{"userId"};
    # See if this username exists in this db
    my ($oldUserId)
        = $db->selectrow_array(q{select userId from users where username="$username"});
    next unless $oldUserId;
    print "Found overlap for user '$username' - old ID: $oldUserId - new ID: $newUserId\n";
    # Now we have an old user ID and a new user ID
```

Fix User IDs

```
#!/data/wre/prereqs/perl/bin/perl
use strict; use DBI;

my $main = DBI->connect("DBI:mysql:dbname=www_plaingui_com","root","password",{RaiseError=>1});
my $merge = DBI->connect("DBI:mysql:dbname=www_webgui_org","root","password",{RaiseError=>1});

# Get all the users from the main db
my $users = $main->selectall_hashref("select username,userId from users","username");
for my $username (keys %{$users}) {
    my $newUserId = $users->{$username}->{"userId"};
    # See if this username exists in this db
    my ($oldUserId)
        = $db->selectrow_array(q{select userId from users where username="$username"});
    next unless $oldUserId;
    print "Found overlap for user '$username' - old ID: $oldUserId - new ID: $newUserId\n";
    # Now we have an old user ID and a new user ID
```

Fix User IDs

```
# Get all the users from the main db
my $users = $main->selectall_hashref("select username,userId from users","username");
for my $username (keys %{$users}) {
    my $newUserId = $users->{$username}->{"userId"};
    # See if this username exists in this db
    my ($oldUserId)
        = $db->selectrow_array(q{select userId from users where username="$username"});
    next unless $oldUserId;
    print "Found overlap for user '$username' - old ID: $oldUserId - new ID: $newUserId\n";
    # Now we have an old user ID and a new user ID
    $db->do(qq{UPDATE $_ SET userId="$newUserId" WHERE userId="$oldUserId"})
        for (qw{ users userProfileData groupings authentication });
    }
}
```

Fix User IDs

```
# Get all the users from the main db
my $users = $main->selectall_hashref("select username,userId from users","username");
for my $username (keys %{$users}) {
    my $newUserId = $users->{$username}->{"userId"};
    # See if this username exists in this db
    my ($oldUserId)
        = $db->selectrow_array(q{select userId from users where username="$username"});
    next unless $oldUserId;
    print "Found overlap for user '$username' - old ID: $oldUserId - new ID: $newUserId\n";
    # Now we have an old user ID and a new user ID
    $db->do(qq{UPDATE $_ SET userId="$newUserId" WHERE userId="$oldUserId"})
        for (qw{ users userProfileData groupings authentication });
    }
}
```

Fix User IDs

```
# Get all the users from the main db
my $users = $main->selectall_hashref("select username,userId from users","username");
for my $user (values %{$users}) {
    my $newUserId = $user->{"userId"};
    # See if this username exists in this db
    my ($oldUserId)
        = $db->selectrow_array(q{select userId from users where username="$username"});
    next unless $oldUserId;
    print "Found overlap for user '$username' - old ID: $oldUserId - new ID: $newUserId\n";
    # Now we have an old user ID and a new user ID
    $db->do(qq{UPDATE $_ SET userId="$newUserId" WHERE userId="$oldUserId"})
        for (qw{ users userProfileData groupings authentication });
}
}
```

Merging Assets

- [Asset IDs are GUIDs (the same improbability applies)

- [Asset Lineage is the main problem

- [`WebGUI/sbin/rebuildLineage.pl`

- Rebuilds the lineage from the asset's parentId

- Fixes any problems we might introduce by our next step

Everything Else

— [Uses GUIDs for the sensitive bits

— [GUIDs are assumed to be unique

The Final Act

Merge the files

```
cp -r /data/domains/www.webgui.org /data/domains/www.plaingui.com
```

Merge the database

```
mysqldump -u<> -p<> --insert-ignore --skip-add-drop-table www_webgui_org | mysql -u<> -p<>  
www_plaingui_com
```

Rebuild the lineage

```
cd /data/WebGUI/sbin  
perl rebuildLineage.pl --configFile=www.plaingui.com.conf
```


The Final Act

— [Clear the Cache

```
rm -rf /tmp/WebGUICache
```

— [Restart and test

- Pay attention to Navigation assets, they may need to be adjusted
- Once everything is working properly, we can exchange our merged site for one of our old sites if desired.

Wrapping Up

Conclusion

- [Split WebGUI sites

- Two separate sites by duplicating and cutting down

- Two virtual sites sharing the same files/database

- [Merge WebGUI sites

- Making one where once was two

Questions?