

Fermilab Central Web Service  
Content Editor Manual

DocDB: CS-doc-5374

## Table of Contents

DocDB: CS-doc-5374 .....	1
1. Role Definitions.....	3
2. Content Editor Role.....	3
3. Tier1 websites and Tier2 websites.....	3
4. Checking existing content editors.....	4
5. Accessing your web content .....	4
6. Web URLs .....	4
7. Where does my content go? / What is each directory or file used for?.....	5
7.1 Default files .....	5
7.2 Default directories .....	5
7.3 Moving and copying files between directories .....	6
8. View your web traffic.....	7
9. Advanced topics.....	7
9.1 Restoring Lost Content.....	7
9.2 Requesting for new scripting languages .....	9
9.3 Requesting for a database .....	10
9.4 Running cron jobs .....	10
9.5 Restarting the Apache HTTPD process.....	10
9.6 Writing web content from a remote machine.....	10
9.7 Mounting another file system to your web server .....	10
9.8 Using chmod, nfs4_setfacl, and other permissions-changing commands.....	10

## 1. Role Definitions

**Site Owner** – Site Owners are the final authority for the website they manage. They manage the content of the website and controls who can edit the web content. Each website should have a primary owner and a secondary owner.

**Content Editor** – A Content Editor is a person who can edit the content of a site. Content Editors have the full permission to change everything within a website. Please note there is no read-only access.

**Service Owner** – The Service Owner is the Web Systems Administration Group. They are responsible for managing the web infrastructure that underpins your website.

## 2. Content Editor Role

The Site Owner is responsible for managing every aspect of their website, including adding and removing Content Editors to the website. As a Content Editor, you have been granted access to edit all aspects of website content. This manual should be used as your guide on how to gain access to the website, how the infrastructure of the website functions, and where you can go for questions on the function of the website.

## 3. Tier1 websites and Tier2 websites

Depending on what Tier of website the Site Owner requested, you will have a few different sets of options available to you when editing a website.

Tier 2 websites enjoy more advanced features and disk space than Tier1 websites. Refer to the following table for a detailed comparison.

<b>Tier 1 website hosting</b>	<b>Tier 2 website hosting</b>
Red Hat Enterprise Linux (RHEL) OS version 6.x	Same
Apache HTTPD web server	Same
Content stored on an NFSv4 file system	Same
Content served over ports 80 & 443	Same
Access to Perl, Python, PHP, and other programming languages available within the RHEL yum repository	Same
Support Availability: 8 x 5	Same

Access to Development and/or Integration websites for your URL	Same
Initial quota of 10GB, up to a maximum of 20GB at no additional cost.	Initial quota of 20GB, up to a maximum of 50GB at no additional cost.
Website is located on a pair of virtual servers, sharing CPU and memory resources with no more than 200 other websites.	Website is located on a pair of virtual servers, with all CPU and memory resources allocated to your website(s).
Offering Cost: None	Offering Cost: One-half of the current rate of two virtual Linux servers, paid annually at the start of each fiscal year
This function is not available for Tier 1 web hosting.	Access to cron on the web server
This function is not available for Tier 1 web hosting.	Direct ssh access to the web server using a local shared account
This function is not available for Tier 1 web hosting.	Access, via the "sudo" command, to restart the Apache HTTPD process if needed.

#### 4. Checking existing content editors

To see who can make changes to your website, navigate to your website via SMB/CIFS or on FNALU and look at the content of the .k5login file. See Section 5 on how to access your web content. Also, .k5login and other important files are described in detail in Section 7.

#### 5. Accessing your web content

Instructions on how to access your web content can be found in this document:

- DocDB Article: CS-doc-5375
- <http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=5375>

#### 6. Web URLs

To access your website from a browser, enter the following URL: <http://YOURWEBSITE.fnal.gov>

For example, if you have a production website at <http://abc.fnal.gov/> and you also requested an associated development site and/or an integration site, their URLs will be

- Development site: <http://abcdev.fnal.gov>
- Integration site: <http://abcint.fnal.gov>.

If you want to use SSL with your website, it is already active. You can access it at <https://YOURWEBSITE.fnal.gov/>

NOTE: Users cannot use just `http://abc/` to access your website. The previous version of the Central Web Service allowed this at one point in time, but it has been found to be a security risk. Therefore the fully qualified name (i.e., <http://YOURWEBSITE.fnal.gov>) must be used.

## 7. Where does my content go? / What is each directory or file used for?

Each website comes with a set of default files and directories. These files and directories are essential to you website, therefore they are configured in such a way that site owners and content editors cannot delete, move, rename or otherwise manipulate them.

### 7.1 Default files

#### **.awstats**

- This file is used to generate web statistics so that you can view your web traffic at <http://YOUR-SITE-NAME.fnal.gov/awstats/awstats.pl>. See Section 10 below for more details.
- Site owners and content editors do not have the permission to view this file.

#### **.k5login**

- If you have a Tier 2 website, this is used to allow users to ssh into your web server. Tier 1 websites do not have this option.
- For both Tier 1 and Tier 2 websites, you can see who has the permission to edit your website by looking at the content of this file.
- Site owners and content editors do not have the permission to edit this file.

#### **.siteowners**

- This file contains an XML listing of the primary and secondary site owners.
- This file is used by the Service Owner for administrative purposes.
- Site owners and content editors do not have the permission to edit this file.

#### **.tier#**

- The # represents the tier level of your website.
- This file is used by the Service Owner for administrative purposes.
- Site owners and content editors do not have the permission to view this file.

### 7.2 Default directories

htdocs / htdocs-dev / htdocs-int

- Put all your production HTML files into the “htdocs” directory. If you have PHP code, it should also be placed in this directory.
- If you have a Tier 2 website, your web scripts can write directly into this directory. If you have a Tier 1 website, then you must use the “data” directory instead.
  - “htdocs-dev” is used for your development site.
  - “htdocs-int” is used for your integration site.

cgi-bin / cgi-bin-dev / cgi-bin-int

- Put your production Perl, Python or other scripts into the “cgi-bin” directory.
- Do not put your PHP code into this directory as it will not work.
  - “cgi-bin-dev” is used for your development site.
  - “cgi-bin-int” is used for your integration site.

data / data-dev / data-int

- Along with the “htdocs” & “cgi-in” directories, these are the only other directories that the web server can see.
- For all Tiers of websites, you can write code to write data into this directory. You can also copy files from this directory to another directory as needed. However you must not move a file to another directory, including “drag-n-drop” or using “CTRL-X and CTRL-P”. Otherwise, you will not be able to see it. Instead, you should copy the file to its new location and delete the original. See Section 9.3 for more details.
- “data-dev” is used for your development site.
- “data-int” is used for your integration site.

### 7.3 Moving and copying files between directories

The directories of your website are configured to protect it against hackers that might want to deface your content. This is done by setting specifically configured ACL’s (Access Control Lists) on each of the directories. These ACL’s manage where the Apache HTTPD server can read content from and where it can write content to. They also block the ability of a hacker to become root on the Linux webserver and delete all your content. This means that you must handle files differently than when you are dealing with them on your local file system.

#### *Why COPY instead of MOVE?*

To get a file from location A to location B within the file system, you must copy it between locations. By using the copy and paste commands, you create a new version of the file in the new location. This allows it to inherit the ACL’s of the directory it was placed in. If you move the file, the Operating System keeps the original ACL’s in-tact, and drops the file into the new location without updating the ACL’s. This is the normal behavior of all Operating Systems.

Here is an example of what can happen if you move a file instead of doing a copy and paste. Assume that you create an index.html file on your desktop and then copy it over to the root directory of your webserver. You then realize you put it in the wrong location and do a drag-n-drop into the “htdocs” directory. Unfortunately, this will break your website. The webserver is not allowed to read files in the root directory of your webserver, and those are the permissions that were applied to that file. When you move it to the “htdocs” directory, the Apache server is unable to read the index.html file because the original ACL’s are still being applied to the file. This will cause your visitors to get a “403 Access Denied” error when visiting your website.

*I accidentally moved the file, how do I fix it?*

The fix is pretty easy. Copy the file(s) to your local machine and delete the problem file(s) on the web server. Now copy the file from your local machine to its intended directory on the webserver. This will create a new version of the file and reset the ACL’s to what they should be.

## 8. View your web traffic

You can view your web traffic at <http://YOURWEBSITE.fnal.gov/awstats/awstats.pl>. Please note that this URL only works when you are on the 131.225.0.0 and 2620:6a:0::0 subnets, or if you are connected to the Fermilab network via VPN.

## 9. Advanced topics

### 9.1 Restoring Lost Content

All of your website contents are backed up once a day. You can access the most recent seven days’ backup yourself.

NOTE: Only the previous seven days’ content are kept online; anything older than that has to be restored from tape.

To restore your content from the previous seven days’ backup, do the following.

*If you use a Linux computer*

1. “ssh” to either FNALU or your webserver (Tier 2 websites only).
2. “cd” to **/web/~snapshot/**.  
NOTE: The “TAB autocomplete” keyboard technique will not work due to the ~ in the directory name; this is normal behavior.
3. Look for directories with names like **####-##-##\_####-####.filesrv01**.
4. Identify the one with the most recent date and “cd” into that directory.
5. From there, navigate into **/sites/YOUR\_WEBSITE/** and look for the file(s) you need to recover.

6. When you find the file(s) you need to recover, copy them directly to where they need to be. For example, if you are in this directory  
`/web/~snapshot/NDMP_AUTO_SNAPSHOT4482/sites/tele.fnal.gov/htdocs`

And you need to restore "index.php", you should use this command:

```
cp index.php /web/sites/tele.fnal.gov/htdocs/
```

NOTE: If you cannot find the file you need in the #####-##-##\_#####-#####.filesrv01 directory with the most recent date, try looking in the other directories with the earlier dates. If you cannot find it there either, then you need to submit a Service Desk request to have it restored from tape. Be sure to include details on the last known date/time of when the file was working.

#### *If you use a Windows computer*

NOTE: These instructions will also work from the general purpose Window Terminal Server "FERMI-TS". If you use a Mac or a Linux computer, you can use the following procedures by connecting to the "FERMI-TS" windows terminal server.

1. Open up Windows Explorer (*WIN-E*) and on the left-side panel, click on the drive where you have `\\filesrv01\web` mounted. For these instructions, we assume it was mounted to "**W:**".
2. At the top of the Windows Explorer window, click in the address bar, and after the "**W:**" type in "~snapshot" so the line looks like this, "**W:**~*snaphot*", and then press return on your keyboard.
3. Review the directory contents and look for the directories with names like #####-##-##\_#####-#####.filesrv01.
4. Identify the one with the most recent date and navigate into that directory.
5. From there, navigate into /sites/YOUR\_URL/ and look for the file(s) you need to recover.
6. When you find the file(s) you need to recover, copy them directly to where they need to be. If you press WIN-E again, it will open a new window, allowing you copy and paste files between these two windows.

NOTE: You cannot drag-n-drop files between these windows. You must use the keyboard shortcuts CTRL-C (copy) and CTRL-P (paste) to restore your files.

7. If you cannot find the file you need in the #####-##-##\_#####-#####.filesrv01 directory with the most recent date, try looking in the other directories with the earlier dates. If you cannot find it there, then you need to submit a Service Desk request so that it can be restored from tape.

#### *If you use a Mac*

**NOTE: While we provide the Mac procedures below, we strongly recommend that you use either the Linux or Windows methods described above as the process is far easier and requires**

**no modifications to your machine. Proceeding to use these instructions will be done at your OWN RISK and without the support of the Fermilab Service Desk.**

Due to the default configurations of the OS X interface, you must modify your Mac's default settings via the command line in order to see the directories needed to restore your files.

1. Open a terminal and enter one of these commands.
  - OS X 10.6 - 10.8:  
% defaults write com.apple.finder AppleShowAllFiles TRUE
  - OS X 10.8 - 10.9:  
% defaults write com.apple.Finder AppleShowAllFiles TRUE
2. Kill your Finder App.
  - killall Finder
3. Start the Finder App and navigate to the **filesrv01.fnal.gov** mount, then to "*web -> ~snapshot*".
4. Look for the directories with names like **####-##-##\_####-####.filesrv01**.
5. Identify the one with the most recent date and navigate into that directory.
6. From there, navigate into **/sites/YOUR\_WEBSITE/** and look for the file(s) you need to recover.
7. When you find the file(s) you need to recover, copy them to where they need to be. If you press CMD-N while in the Finder window, it will open a new window, allowing you to copy and paste files between these two windows.  
NOTE: You cannot drag-n-drop the files between the windows. You must use the keyboard shortcuts CMD-C (copy) and CMD-P (paste) to restore your files.
8. If you cannot find the file you need in the **####-##-##\_####-####.filesrv01** directory with the most recent date, try looking in the other directories with the earlier dates. If you cannot find it there, then you need to submit a Service Desk request so that it can be restored from the tape.
9. To reverse your changes to the OS, do the following:
  - a. Open a terminal and enter one of these commands.
    - OS X 10.6 - 10.8:  
defaults write com.apple.finder AppleShowAllFiles FALSE
    - OS X 10.8 - 10.9:  
defaults write com.apple.Finder AppleShowAllFiles FALSE
  - b. Kill your Finder App by entering the following command:
    - killall Finder

## 9.2 Requesting for new scripting languages

By default, you should have access to Perl, Python, and PHP. You can submit a Service Desk request if you would like to use another scripting language. The Service Owner will install it for you if the requested scripting language is in the default YUM software library and it does not violate any computing security policies.

### 9.3 Using a Database with your web server

You can submit a Service Desk request if you need MySQL or another database to run your web application. Your request will be routed to the Database Administration Group for processing.

### 9.4 Running cron jobs

This feature is only available to site owners and content editors of Tier 2 websites. To create a cron job, ssh into your web server. When you are logged in via the shared user, you can create a cron entry for that user.

### 9.5 Restarting the Apache HTTPD process

This option is only available to site owners and content editors of Tier 2 websites. If this is a function you require, you can submit a Service Desk request and the Central Web Hosting Service Owner will discuss the matter with you.

### 9.6 Writing web content from a remote machine

If you need to write data to your web directory from a remote machine, you will need a special Kerberos principal. Contact the Service Desk for help with obtaining that.

When you receive the special principal from the Service Desk, you can submit a Service Desk request, asking the Central Web Hosting Service Owner to add the principal to the list of content editors for your website, thereby granting it permission to write files into the file system for your website. You can then mount the file system to your remote machine and run your processes.

### 9.7 Mounting another file system to your web server

If you want to mount another file system to your web server, please submit a Service Desk request and the Central Web Hosting Service Owner will discuss the matter with you.

### 9.8 Using `chmod`, `nfs4_setfacl`, and other permissions-changing commands

The file system has been locked against all permissions changes. You will be able to run the commands which will change the permissions, but nothing will happen. If you try the permissions changes on from a Windows machine, it will report back that you do not have permission to make the changes.

### 9.9 Restricting Access to your Production Website

To limit who has access to your website is a simple matter. Create a file called “.htaccess” (note the period at the front of the file) and put this content in said file.

```
order deny,allow
deny from all
allow from 131.225
allow from 2620:6a:0::/48
```

This will limit all access to your website to only the Fermilab-owned subnets. This file covers both IPv4 and IPv6.

For more advanced ways to limit access, a simple Google search for “apache .htaccess examples” will give you a wealth of information.