This page has some background information on making backups and explains some basic *nix backup and restore procedures.

Introduction

Your wiki installation contains some unique data in the following directories:

`local/ Local configuration scripts cookbook/ Recipes obtained from the Cookbook pub/ Publicly accessible files wiki.d/ Wiki pages uploads/ Uploaded files (attachments)`

A good backup plan will include periodically archiving these directories -- or at bare minimum `local/` and `wiki.d/`. Good practice dictates keeping your backup archives on a separate machine.

Simple Backup and Restore (*nix)

When it comes to backup, simpler is better.
Since the pmwiki distribution is very small (about 1/4 megabyte), it's simplest to just archive the distribution files along with the data.

Making a Backup Archive

The following *nix command, executed from the parent directory of your wiki's directory, will put a complete backup archive of your site in your home directory.

```
 tar -zcvf ~/wiki-backup-`date +%Y%m`.tar.gz wiki/
```

Restoring the Backup Archive

Simple Method

Your site can be restored and running in under 30 seconds with

```
 tar -zxvf ~/wiki-backup-200512.tar.gz find wiki/uploads/ -type d |xargs chmod
777 find wiki/wiki.d/ -type d |xargs chmod 777
```

A Slightly-More-Secure Method

The simple restore commands above will give you world-writable files and directories. You can avoid world-writable permissions by letting Pm Wiki create directories with the proper attributes (ownership and permissions) for you.

Start with

```
 tar -zxvf ~/wiki-backup-200512.tar.gz rm -rf wiki/wiki.d rm -rf uploads chmod 2777 wiki/
```

Now upload a file in each group that had uploads. If your site doesn't have uploads, just visit your site once so the wiki.d/ directory will be created.

Finish your installation with

```
 chmod 755 wiki/ tar -zxvf ~/wiki-backup-200512.tar.gz
```

Details

The commands on this page assume your site is in a directory called "wiki/". The test backup was made in December, 2005 so it's named accordingly.

Your site will only have an uploads/ directory if uploads are enabled.

The backup command uses a date stamp (YYYYMM) in the filename. If you automate the command via cron you'll wind up with monthly snapshots of your site. You can get a daily snapshot by appending %d to the date command (`` `date +%Y%m%d` `` will get you YYYYMMDD). Be wary of space limitations if you have a large uploads/ directory.

See Also

- A thread [gmane.org] on the pmwiki-users mailing list.
- A Backup Pages recipe in the cookbook.

Miscellaneous

Backup via FTP

Download and install a ftp client like Filezilla

1. Using the ftp client connect to the server where you host pmWiki using
   1. the IP address (ex: 123.234.56.67) or the ftp name (ex: ftp.myhost.com)
   2. supply your account name (ex: mylogin) and password (ex: myp4ssw0rd)
2. Move to your pmWiki directory (ex: `/usr/mylogin/web/wiki/` or `/tahi/public_html/pmwiki`)

3. Select the folder you want to backup as explained before (probably either only the data or the whole wiki directory)
   - for data you will want to backup both the directories
     - `wiki.d` for user page data
     - `pmwikiuploads` (or `uploads`) for your attachments (uploads)
   - for system you will want, at a minimum, to backup both the directories
     - `local` for configuration data
     - `pub` for local CSS and skins customisations
4. Download them to a local folder
5. Use 7zip or a similar software to build an archive of this backup

You can also very easily sync your FTP directories with your hard disc via this command line:

```
wget -nv -np -m ftp://user:password@ftp.yourhost.net/
```

Download Wget for Windows (other systems normally have it installed).

Alternatively, you can also mirror your FTP directories with lftp:

```
lftp -u your_user_name,your_password -e "mirror --verbose /wiki.d
/path/to/local/folder" ftp://your_host
```

(this will mirror only the /wiki.d folder, replace with / to mirror everything)

Using rsync

See Cookbook:BackupWithRsync and Cookbook:TwoWayMirroringWithRsync.