

Lesson 3 Outline - Script Installation

- I. What should you know about your ISP?
 - A. CGI directory access?
 - B. What OS is the server using?
 - C. Do you have shell (telnet) access or FTP only?
 - D. SSI support?
 - E. What are the paths to perl, sendmail, etc?
 - F. Find the ISP's FAQ for answers to most of these questions

- II. Uploading the script
 - A. The script usually comes in compressed format; you will need the appropriate utility to uncompress and/or unpack it (*.gz, *.Z, *.tar, *.zip)
 - B. It's a lot easier to uncompress and unpack on the web server rather than uploading individual files. Also, a tar'd file usually has the correct file permissions already set.
 - C. Upload files in the proper mode to avoid corruption
 1. Use binary (a.k.a. raw data) for compressed files and images ONLY
 2. Use text (a.k.a. ASCII) for text files and scripts
 - D. You will then need to put things into the right place on the web server. Refer to the script's README for directions.
 1. Put CGI scripts into the cgi-bin (per server configuration)
 2. Some ISPs will not allow HTML files inside the cgi-bin directory
 3. Sensitive files (ex. Password files or order logs) should be outside of the web accessible directory tree if possible

- III. Setting file permissions
 1. Generally, the script will have an accompanying README file that will instruct you on setting the appropriate file permissions
 2. On Windows NT, you must contact tech support to set permissions (unless you have PCAnywhere or access to the server directly)
 3. On UNIX, use chmod to set proper permissions via telnet, or use your FTP program
 4. All CGI scripts must be executable by the web user (usually "nobody")
 5. Other perl files (ex. Libraries) only need to be readable
 6. Any files to be read by the CGI script must be readable by the web user
 7. Any files to be written by the CGI script must be writeable by the web user
 8. If the script will be creating new files, the *directory* must be writeable by the web user
 4. It's important to know who the web user is on the system (refer to Lesson 2 outline for explanation). If the web user is "nobody", as is typically the case, script permissions must be set for "everyone" (a.k.a. world-writeable or world-readable).
 1. File world-executable: 755 (rwxr-xr-x)
 2. File world-writeable: 666 (rw-rw-rw-)
 3. File world-readable: 644 (rw-r--r--)
 4. Directory world-writeable: 777 (rwxrwxrwx)
 5. You shouldn't grant more permissions than is necessary (for security reasons)
 6. On UNIX, the user you use to FTP files is generally not the same as the web user. This can cause confusion with file permissions.
 1. If you upload a file via FTP, it will typically have write permission only for you; if the script needs permissions other than read-only, you will likely have to set them manually.
 2. If the script creates a file, you may not have permission to view it (or modify it) from FTP. This is because the web user owns the file, not you. There are a few ways to handle this.
 - a. If you know what files the script will be creating, you can create them yourself before running the script. That makes you the owner, not the web user. To create an

empty file use the UNIX “touch” command. Then you will need to make the file writable by the web user.

- b. If you have telnet access, you can usually gain ownership by renaming the file to another name, and then copying it back to the original name (Ex. “mv origfile origfile.save; cp origfile.save origfile”)
- c. Edit the script to set the file permissions to world-readable after the file is created.
- d. You can try using the nobody.cgi script (or write your own) to execute a UNIX chown (change owner) or chmod command from a web script. **DO THIS WITH CAUTION!** Giving shell access to a web script is a BIG security risk. Delete the script when you’re done with it.
- e. Contact tech support and ask them to chown the file for you.

IV. Initializing variables

- A. Probably one of the most confusing points is knowing the difference between a “path” and a “URL”
 1. Use relative paths if possible to avoid dependency on the ISP’s setup (and sometimes you don’t actually know the full path)
 2. Use absolute URLs for simplicity – remember that relative URLs must be based on the location of the script itself, not base web directory
- B. Often you must supply the path to system resources such as sendmail or perl. Refer to your ISP’s FAQ page or use the UNIX shell to determine the paths.
- C. Remember that you must escape the @ in email addresses when used in perl strings (ashley\@enscript.com)

V. Script portability

- A. There are several perl functions that do not port from UNIX to Windows NT
 1. system()
 2. `command` notation
- B. Remember the differences between UNIX and Windows directory and file naming conventions
- C. Sendmail is not typically available on NT systems. You will have to use an alternate method for sending email from a script. Make sure that the script supports this.
 1. SMTP server
 2. Blat
 3. WindMail