

Duke Office of Information Technology

CGI.DUKE.EDU

cgi services for the Duke Community

cgi.duke.edu

The cgi.duke.edu server is available for acceptable use by Duke faculty, staff, and students. Notes regarding use of this machine can be found below.

CGI Notes

1. For a user to access the cgi server, run `/usr/local/bin/create-cgikey`
2. The [Service Desk](#) can create cgi keys for projects and organizations, provided the requests are made by the ones who administer those pages.
3. By creating a cgi key and running scripts on cgi.duke.edu, users are agreeing to the [usage policy](#)
4. In order to execute, all CGI scripts must end with either `.pl` or `.cgi` extensions. Any other files, images, etc., ending with these extensions will generate an error. A `cgi-bin` directory is not necessary. The files may reside in any web accessible directory.
5. A few things to be aware of when writing cgi scripts...
In order for your script to write to a directory, or insert a file, the permissions should be set to `rliw` for the `cgi.user pts id` or `cgi.project_projectname pts id`.

?

fs sa directory cgi.user rliw

? The command syntax to use would be:

It is recommended that you create a special `cgi-insert` or `insert` subdirectory so that files do not get overwritten or otherwise damaged by your cgi scripts.

`FORM METHOD=get ACTION="/some/path/to/script.cgi"> <FORM METHOD=get ACTION="http://cgi.duke.edu/~userid/some/path/to/script.cgi">` In writing forms, the action specified for GET methods can be referenced either relatively or absolutely, for example: OR HOWEVER, the action specified for POST methods need to be referenced absolutely for the time being: `<FORM METHOD=post ACTION="http://cgi.duke.edu/~userid/some/path/toscript.cgi">`

You are responsible for error and limit checking within your cgi scripts.

- 6.
7. CGI scripts, unlike ASPs and ASP-like active content, must be executed by the CGI server, and since the CGI server is a Unix machine, scripts must be flagged as executable within the filesystem in order to work. Most text files, when created, don't get their execute bits enabled by default, so users will typically need to enable execute permission on a file-by-file basis for their individual CGI scripts. The command syntax to use is:
`chmod +x filename`

- 8.
9. Using PHP in CGI scripts raises some additional concerns. As with any CGI script, a PHP-interpreted CGI script is responsible for emitting appropriate HTTP headers -- the web server will not automatically generate HTTP headers for the output of CGI scripts. The minimal valid HTTP header consists of a single line containing a MIME-style "Content-Type:" header followed by a blank line, emitted as the first two lines output by the script. PHP users of the CGI server will need to make sure that their scripts emit proper HTTP headers, just like other CGI scripts. A typical beginning for a valid PHP CGI script might look like this:

```
#!/usr/local/bin/php Content-Type: text/html ...rest of PHP script...
```

PHP users should also note that the PHP interpreter used on our CGI server runs in "CLI" mode, which among other things means that certain PHP routines (eg., *phpinfo()*) will emit preformatted plain text rather than HTML. In general, the CGI PHP interpreter doesn't "know" that it's running in a web environment and as such, won't automatically format its output in HTML. PHP CGI scripts need to be written to take this into account.

On this page

- ? The user refers to *your* ACPUB userid or ACPUB login. For example, if your userid is *hiro*, your cgi pts id would be:
 - ? *cgi.hiro*
- ? If your organization, FOO, had web pages at <http://www.duke.edu/web/foo>, the cgi pts id would look like:
 - ? *cgi.project_foo*

Duke Office of Information Technology - www.oit.duke.edu - (919) 684-2200 - help@oit.duke.edu