

Poor man's comments: Inserting text that has no effect into a configuration file

devblogs.microsoft.com/oldnewthing/20140917-00

September 17, 2014



Raymond Chen

Consider a program which has a configuration file, but the configuration file format does not have provisions for comments. Maybe the program has a “list of authorized users”, where each line takes the form `allow x` or `deny x`, where `x` is a group or user. For example, suppose we have `access_list` that goes like this:

```
allow payroll_department
deny alice
allow personnel_department
allow bob
```

This is the sort of file that can really use comments because people are going to want to know things like “Why does Bob have access?”

One way of doing this is to embed the comments in the configuration file in a way that has no net effect. You can do this to add separator lines, too.

```
deny !_____
allow payroll_department
deny !alice_is_an_intern_and_does_not_need_access_to_this_database
deny alice
deny !_____
allow personnel_department
deny !_____
deny !temporary_access_for_auditor
deny !see_service_request_31415
deny !access_expires_on_2001_12_31
allow bob
```

Assuming that you don’t have any users whose names begin with an exclamation point, the extra `deny !...` lines have no effect: They tell the system to deny access to a nonexistent user.

Sometimes finding the format of a line that has no effect can take some creativity. For example, if you have a firewall configuration file, you might use URLs that correspond to no valid site.

```
allow nobody http://example.com/PAYROLL_DEPARTMENT/-----
allow alice http://contoso.com/payroll/
allow nobody http://example.com/PURCHASING_DEPARTMENT/-----
allow bob http://contoso.com/purchasing/
allow nobody http://example.com/SPECIAL_REQUEST/-----
allow ceo https://www.youtube.com/
```

Of course, these extra lines create work for the program, since it will sit there evaluating rules that will never apply. You may have to craft them in a way so that they have minimum cost. In the example above, we assigned the comments to a user called `nobody` which presumably will never try to access the Internet. We definitely didn't want to write the comment like

```
allow * http://example.com/PAYROLL_DEPARTMENT/-----
```

because that would evaluate the dummy rule for every user.

If you are willing to add a layer of process, you can tell everybody to stop editing the configuration files directly and instead edit an alternate file that gets preprocessed into a configuration file. For example, we might have `access_list.commented` that goes

```
////////////////////////////////////
allow payroll_department
deny alice // payroll intern does not need access to this database.
////////////////////////////////////
allow personnel_department
////////////////////////////////////
allow bob // Temporary access for auditor, see SR 31415. Expires 2001/12/31.
```

Everybody agrees to edit the `access_list.commented` file, and after each edit they run a script that sends the file through the C++ preprocessor and puts the result in the `access_list` file. By using the C++ preprocessor, you enable features like `#include` directives and `#define` macros.

Raymond Chen

Follow

