

Re: 2 simple shell related questions

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.misc/2007-10/msg00755.html>

- *From:* "Chris F.A. Johnson" <cfajohnson@xxxxxxxxxx>
 - *Date:* Wed, 10 Oct 2007 21:02:06 -0400
-

On 2007-10-11, babaji wrote:

Newbie alert!

Hello everyone :) . I have two simple questions regarding shell variables. I tried looking for the answers on the net but couldn't find an exact match.

Q1: what does \$^ represent

Q2: what does \$@ represent

I saw both in a makefile and searched the GNU make document which has references to these symbols but does not explain clearly what they stand for (or maybe I didn't get it!) . I searched on google too but came up with not so clear results.

From what I know \$ is used to get the value of a shell var,
and so ^

or @ must stand for something like a special-var.

I know what \$#, \$0/1/2... and \$* are used for but can't get an idea about these two :)

They are not shell variables; a makefile is not a shell script. See the documentation for make:

Automatic Variables

`\$@'

The file name of the target of the rule. If the target is an archive member, then `\$@' is the name of the archive file. In a pattern rule that has multiple targets (*note Introduction to Pattern Rules: Pattern Intro.), `\$@' is the name of whichever target caused the rule's commands to be run.

Re: 2 simple shell related questions

`\$^`

The names of all the prerequisites, with spaces between them. For prerequisites which are archive members, only the member named is used (*note Archives:). A target has only one prerequisite on each other file it depends on, no matter how many times each file is listed as a prerequisite. So if you list a prerequisite more than once for a target, the value of `\$^` contains just one copy of the name.

—

Chris F.A. Johnson, author | <<http://cfaj.freeshell.org>>
Shell Scripting Recipes: | My code in this post, if any,
A Problem–Solution Approach | is released under the
2005, Apress | GNU General Public Licence

.