
Subject: Re: Checking for perl
Posted by [David Fanning](#) on Wed, 07 Mar 2007 14:46:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Panter writes:

> My code runs on linux with spawn quite happily, but I'd really like
> to also have it run in windows. On my xp laptop this is fine - I have
> perl set up, and it's happy to be spawned. The problem comes, of course,
> when I give the code to someone else. I'd like to be able to safely
> check whether perl is installed, and if not return a *device
> independent* warning that the action wasn't possible - without crashing
> the code.... any ideas?

Maybe you could search the PATH environment variable for "perl":

```
IDL> thePath = GetEnv('PATH')
IDL> IF StrPos(StrUpCase(thePath), 'PERL\BIN') NE -1 $
    THEN Print, 'PERL is allowed'
```

PERL is allowed

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Checking for perl
Posted by [Ben Panter](#) on Wed, 07 Mar 2007 15:03:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Maybe you could search the PATH environment variable for "perl":
>
> IDL> thePath = GetEnv('PATH')
> IDL> IF StrPos(StrUpCase(thePath), 'PERL\BIN') NE -1 \$
> THEN Print, 'PERL is allowed'
>
> PERL is allowed
>
> Cheers,
>

> David

Thanks David - unfortunately that fails under linux when perl is aliased rather than in the path. I think I probably just have to accept that I need to write a multi-os code.

Cheers,

Ben

--

Ben Panter, Edinburgh, UK.
Email false, <http://www.benpanter.co.uk>
or you could try ben at ^^^^^^^^^^^^^^^

Subject: Re: Checking for perl
Posted by [David Fanning](#) on Wed, 07 Mar 2007 15:18:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Panter writes:

> Thanks David - unfortunately that fails under linux when perl is aliased
> rather than in the path. I think I probably just have to accept that I
> need to write a multi-os code.

Well, I would assume so. But !VERSION is always good for this. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Checking for perl
Posted by wlandsman@jhu.edu on Wed, 07 Mar 2007 15:22:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

"> >

> command="perl -e 's/%%Title: Graphics produced by IDL/%%Title:
> "+new_title+"/" -p " + filename + " > bp_pstitle.temp"
> command2="rm "+filename

```
> command3="mv bp_pstitle.temp " + filename
>
> spawn, command
> spawn, command2
> spawn, command3
>
```

As a side note, you could replace the second and third spawn commands with the OS-independent FILE_DELETE and FILE_MOVE commands.

Maybe you can just run a simple test perl program and capture the output from spawn.

For example,

```
spawn,'perl -h',tt
if N_elements(tt) GT 1 then begin
    ;Now do the real perl command
endif else message,'PERL not installed, postscript file not updated'
```

Another kludge would be write a little IDL program that reads the postscript file one line at time, tests for the presence of the "IDL graphics" line and updates it and writes the line to a new postscript file. Or you could read the entire postscript file into an IDL string array, and then do a global replace on the offending text.

--Wayne

Subject: Re: Checking for perl
Posted by [Ingo von Borstel](#) on Wed, 07 Mar 2007 15:27:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
Hi,
Ben Panter wrote:
> David Fanning wrote:
>
>> Maybe you could search the PATH environment variable for "perl":
>>
>> IDL> thePath = GetEnv('PATH')
>> IDL> IF StrPos(StrUpCase(thePath), 'PERL\BIN') NE -1 $
>>     THEN Print, 'PERL is allowed'
>>
>>     PERL is allowed
>>
>> Cheers,
>>
>> David
>
```

> Thanks David - unfortunately that fails under linux when perl is aliased
> rather than in the path. I think I probably just have to accept that I
> need to write a multi-os code.

You could check for the OS type using the system variable `!version.os_type`
Depending on the result you could implement a check whether it's in the
path (windows) or spawn a 'whereis perl' or evaluate the result of "`ls
-R /usr | grep 'perl' "` (unix / linux). In the latter case, of course,
it might happen, that a whereis command isn't available and the method
using 'grep' may take quite some while.

Maybe that are some ideas to work out?

Best regards,
Ingo

--

Ingo von Borstel <newsgroups@planetmaker.de>
Public Key: <http://www.planetmaker.de/ingo.asc>

If you need an urgent reply, replace newsgroups by vgap.

Subject: Re: Checking for perl
Posted by [Ben Panter](#) on Wed, 07 Mar 2007 15:34:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wayne Landsman wrote:

> spawn,'perl -h',tt
> if N_elements(tt) GT 1 then begin
> ;Now do the real perl command
> endif else message,'PERL not installed, postscript file not updated'

That's exactly what I was after Wayne, and I'm kicking myself for not
thinking of it.

Thank you!

Ben

--

Ben Panter, Edinburgh, UK.
Email false, <http://www.benpanter.co.uk>
or you could try ben at ^^^^^^^^^^^^^^^
