Build and Source Control - Bug #3219

recent Ubuntu/Linux builds use ncurses 6.0 and we don't link to it properly

12/08/2016 06:09 AM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Related to Build and Source Control - Bug #3298: libp2j build links with ncur...	**Closed**

History

#1 - 12/08/2016 06:33 AM - Greg Shah

- Subject changed from recent Ubuntu/Linux builds and ncurses 6.0 to recent Ubuntu/Linux builds use ncurses 6.0 and we don't link to it properly

- File makefile.diff added

In Ubuntu 16.04 (maybe in 15.10, I'm not sure) Ubuntu shifted from NCURSES 5.9 to NCURSES 6.0. As far as I can tell, if you have libncurses.so.5.9 installed and you have upgraded to 16.04, that library is not removed. It will stay installed but it won't be updated.

If that old version was already patched at the time of the upgrade, then it will remain in its patched state forever.

If that old version was not patched at the time of the upgrade, then it remains unpatched forever. Our standard patching process (patch_ncurses.sh) uses the apt-get source ncurses to pull the standard source package for the platform. This means that after the upgrade, it will pull ncurses 6.0 (ncurses_6.0+20160213 as of today). The patching will install a new version of libncurses.so.6, but will not and cannot update libncurses.so.5.9.

The problem is that our native build process links with the default libncurses.so (using the -lncurses linker option). On the system's I've checked, the default always it seems to be the 5.9 version. If your system has the unpatched version, this will fail and no amount of patching the 6.0 version will help. Since you cannot patch the 5.9 version (at least with our standard process), you are "stuck".

To force it to link with 6.0, we can use the attached diff. It is not clear to me if this is the right solution. In particular, we haven't tested 6.0 as much (maybe at all) AND when it changes to 6.1 or 7.0 we will have ot make changes to the build to match.

Other options:

1. Remove the old 5.9 version. I'm not sure if that is safe to do. Nor am I clear on the exact process for doing that, since there are many files involved in that installation and there is configuration too (e.g. ldconfig).

2. Find a different way to patch the 5.9 version. This would have to deliberately pull down the specific version that matches the old install and run the patching using that code. It doesn't seem like a good idea.

3. Change the default library that is used for linking, so that the system links with 6.0 by default.

Thoughts? Other ideas?

Does anyone have a fresh install of 16.04? If so, is there a 5.9 installed there at all?

#2 - 12/08/2016 07:07 AM - Hynek Cihlar

Could we maybe use a different dependency mechanism, like Snappy? So that the system library is left intact and we isolate our patched library only for P2J?

#3 - 12/08/2016 07:53 AM - Greg Shah

The only problem there is that it is pretty specific to Ubuntu. I want to make sure we have a general purpose support for all Linux distros (also our Linux support is generally used for Solaris/UNIX too).

One similar thought to yours is that we could ship our own fork of NCURSES, named something completely different and pre-patched. This could be installed once and could even come with a pre-configured custom terminfo.

We also have #2660, which has the potential to eliminate our need to patch ncurses.

#4 - 09/07/2017 08:03 AM - Greg Shah

- Related to Bug #3298: libp2j build links with ncurses 5.9 when the proper version is 6.0 added

#5 - 09/07/2017 08:04 AM - Greg Shah

- Status changed from New to Closed

This is the same issue as #3298.

Files

makefile.diff	546 Bytes	12/08/2016	Greg Shah