

Build and Source Control - Support #5167

using static linking to eliminate the need to patch the system-wide ncurses

02/26/2021 08:02 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	Yes		
vendor_id:			
Description			
Related issues:			
Related to User Interface - Support #2660: evaluate if the NCURSES 5.7 "threa...			New
Related to Runtime Infrastructure - Support #4549: reduce/eliminate installat...			New
Related to Build and Source Control - Bug #5166: libp2j.so links to unpatched...			Closed

History

#1 - 02/26/2021 08:05 AM - Greg Shah

We would like to eliminate the need to patch the system-wide version of ncurses. [#2660](#) is designed to do exactly that but we do not know if it will work.

If [#2660](#) cannot be made to work OR if we just don't have time to try it, then a fall-back plan is to [statically link](#) the libp2j.so to a patched ncurses. This should avoid the need to install the patched version as the system-wide ncurses.

#2 - 02/26/2021 08:05 AM - Greg Shah

- Related to Support #2660: evaluate if the NCURSES 5.7 "threading improvements" can be made to work for P2J such that `auto_getch_refresh()` is no longer needed added

#3 - 02/26/2021 08:05 AM - Greg Shah

- Related to Support #4549: reduce/eliminate installation dependencies added

#4 - 02/26/2021 09:44 AM - Greg Shah

- Related to Bug #5166: libp2j.so links to unpatched ncurses 6.2 instead of the patched ncurses 5.9 added

#5 - 02/26/2021 09:44 AM - Greg Shah

- Subject changed from `usign static linking` to `eliminate the need to patch the system-wide ncurses to using static linking to eliminate the need to patch the system-wide ncurses`

#6 - 04/27/2023 11:45 AM - Greg Shah

- Assignee set to Eugenie Lyzenko

Please give this idea a try to see if it can work. You would patch a local version of the ncurses project and use it for the linking step in our FWD build, without ever patching the system-wide ncurses installation. If that works, we should give some thought to moving to this approach since it would reduce the burden on system installations.

The downside is that we would need that locally patched version to be available at FWD build time, but I don't think that is harder to achieve than the

system changes themselves, so it would be an overall win.

#7 - 05/01/2023 03:53 PM - Eugenie Lyzenko

Greg Shah wrote:

Please give this idea a try to see if it can work. You would patch a local version of the ncurses project and use it for the linking step in our FWD build, without ever patching the system-wide ncurses installation. If that works, we should give some thought to moving to this approach since it would reduce the burden on system installations.

The downside is that we would need that locally patched version to be available at FWD build time, but I don't think that is harder to achieve than the system changes themselves, so it would be an overall win.

Greg,

The short answer: yes we can do static linkage with ncurses and use it with FWD projects.

But we need to do some changes in FWD native library part:

1. Probably we still need to install regular ncurses dev package to use *.h headers in libp2j.so building of the FWD project: ncurses.h and term.h.
2. We certainly need to include the object libraries in FWD project to do static linkage: libncurses.a and libtinfo.a. Not sure about last one but without it the FWD works strange and not good (probably attempts to use system wide version of libtinfo.so). We need to ensure both libraries to come from same source.
3. The alternative option should be used to configure ncurses before the object libraries to become compatible with static linking: ./configure --with-termlib CFLAGS='-fPIC -O2' --with-abi-version=6. Or we can just provide precompiled binaries for both object libs. Say we can put these libraries in src/native/linux_lib subdirectory of the project.

The last thing to do here is to ensure this will work on clean OS with default(not-patched) ncurses installed. This should ensure compatibility and co-existence of such installations. And this is what I'm going to test next.

#8 - 05/01/2023 09:38 PM - Eugenie Lyzenko

- % Done changed from 0 to 90

- Status changed from New to Feedback

I made the compatibility testing for clean Ubuntu 22.04 system and statically linked patched ncurses calls from libncurses.a and libtinfo.a. The conclusions:

1. This config works fine. Tested with hotel GUI application.
2. To build the libp2j.so on Linux we need 2 include files: ncurses.h and term.h.
3. Ironically these include file are not included in ncurses source code we use to prepare patched object libraries for static link.

So on target system we have to either install libncurses-dev package (not patched) or to include these files in FWD project. The last option is not a good on my guess because the ncurses.h or term.h are designed to be in system include directory and can have references to other *.h files. I think it is safe to install libncurses-dev

The only we need to include in FWD are the object libraries in some directory and make changes in libp2j.so makefile to start using static link. I think we can do this right now. The directory to store the *.a files is the subject of discussion.

When we will think the *.a are need to be updated - we just rebuild new patched objects and update FWD. Honestly I do not think it should be done very often.

#9 - 05/02/2023 07:10 AM - Greg Shah

Don't we already require that libncurses-dev is installed to compile? ([Install NCURSES Development Headers](#))

the object libraries in some directory

Can these be found in the LD_LIBRARY_PATH? If so, we can augment that to point to our custom dir where the patched ncurses exists.

We will need to update our patching process ([Patching NCURSES](#)) to create that directory

Roger: Are the scripts posted there the latest ones we are using in our standard workstation?

#10 - 05/02/2023 07:10 AM - Greg Shah

- Status changed from Feedback to WIP

#11 - 05/02/2023 07:11 AM - Greg Shah

Yes, we will move ahead with this approach now. Let's get the details figured out and you can implement the changes.

#12 - 05/02/2023 08:13 AM - Eugenie Lyzenko

Greg Shah wrote:

Don't we already require that libncurses-dev is installed to compile? ([Install NCURSES Development Headers](#))

Yes we did install it before. I just wanted to state with new approach we can not avoid this step (as I wanted to be). In ideal case the developer should not know anything about ncurses installation requirement, just take FWD repo and build it.

#13 - 05/02/2023 08:21 AM - Eugenie Lyzenko

Greg Shah wrote:

Can these be found in the LD_LIBRARY_PATH? If so, we can augment that to point to our custom dir where the patched ncurses exists.

We will need to update our patching process ([Patching NCURSES](#)) to create that directory

OK. I'll test this way.

Another word you do not want to include *.a files in FWD project tree and link them as the makefile change like this?

```
ifeq "$(OS)" "Linux"
    override CFLAGS+=-fpic
...
    override LDFLAGS+=-L$(SRCDIR)/linux_lib -l:libncurses.a -l:libtinfo.a -ldl -lutil
...
```

#14 - 05/02/2023 08:42 AM - Greg Shah

Another word you do not want to include *.a files in FWD project tree and link them as the makefile change like this?

No. We will assume there is still a special patching step needed for any system on which the FWD native code will be built (linked really). However, this would allow any system using a binary build of FWD to avoid the patching, which is a big improvement.

#15 - 05/02/2023 09:12 AM - Roger Borrello

Greg Shah wrote:

Roger: Are the scripts posted there the latest ones we are using in our standard workstation?

I'm not sure where that post came from. [This post](#) is what I used to validate the gcd_systems ncurses playbook. In it, we explicitly remove libncurses5-dev because we are working with ncurses 6.x.

```
---
# Apply the FWD patches to both NCURSES and TERMINFO.

- name: "Enabling source packages (needed before we try to patch ncurses)."
  replace:
    dest: /etc/apt/sources.list
    regexp: "# deb-src"
    replace: "deb-src"
    backup: yes
    become: yes

- name: "Make sure that the package list is up to date."
  apt:
    update_cache: yes
    become: yes
```

```

- name: "Ensure that libncurses5-dev is not installed."
  apt:
    name: libncurses5-dev
    state: absent
    become: yes

- name: Place ncurses patching files
  copy:
    src: "{{ item.src }}"
    dest: "{{ item.dest }}"
    mode: "{{ item.mode }}"
  become: yes
  loop:
    - { src: "patch_ncurses.sh", dest: "/root/patch_ncurses.sh", mode: "u+x" }
    -
{ src: "ncurses_lib_getch_c_v6.1_20200708.patch", dest: "/root/ncurses_lib_getch_c_v6.1_20200708.patch", mode:
  "" }
    -
{ src: "ncurses_curses_h_in_v6.1_20200708.patch", dest: "/root/ncurses_curses_h_in_v6.1_20200708.patch", mode:
  "" }
    - { src: "patch_terminfo.sh", dest: "/root/patch_terminfo.sh", mode: "u+x" }
    - { src: "manage_term_patches.sh", dest: "/root/manage_term_patches.sh", mode: "u+x" }
    - { src: "95patch-ncurses", dest: "/etc/apt/apt.conf.d/95patch-ncurses", mode: "" }

- name: Perform patching
  command: /root/manage_term_patches.sh
  become: yes

```

The tail end of the patching process is:

```

...
# patch the source code
patch ncurses/base/lib_getch.c $getchpatch
patch include/curses.h.in $headerpatch

# build the source code
./configure --libdir=/usr/lib --with-shared --with-abi-version=6 && make

# install it
sudo make install

# identify the .so name (it is version-specific)
full_libname=$(basename $(find lib/ -maxdepth 1 -type f -name "libncurses.so.*"))
major_libname="${full_libname%.*}"

# fixup libraries permissions to be accessible for building
sudo chmod o+rx /usr/lib/libform.so.* /usr/lib/libmenu.so.* /usr/lib/libncurses.so.* /usr/lib/libpanel.so.*

# fixup the main lib (the install process leaves it alone, but that won't work)
cd /lib/$(uname -i)-linux-gnu/
# don't simply append '.ori' or future apt commands will 'fix' it
sudo cp -p $full_libname "${full_libname/ncurses./ncurses-BEFORE-FWD-PATCH.}.ori"
sudo cp $full_libname $full_libname.ori
sudo cp /usr/lib/$full_libname .
sudo rm $major_libname
sudo ln -s $full_libname $major_libname

cd "$starting_dir"

# cleanup
rm -fr "$srcdir" ncurses_*.tar.gz ncurses_*.tar.xz ncurses_*.dsc ncurses_*.asc

```

#16 - 05/02/2023 10:52 AM - Eugenie Lyzenko

Greg Shah wrote:

Don't we already require that libncurses-dev is installed to compile? ([Install NCURSES Development Headers](#))
the object libraries in some directory

Can these be found in the LD_LIBRARY_PATH? If so, we can augment that to point to our custom dir where the patched ncurses exists.

We will need to update our patching process ([Patching NCURSES](#)) to create that directory

The LD_LIBRARY_PATH does not work. At least without severe OS system wide changes (ldconfig I guess).

But the LIBRARY_PATH works fine. The bash settings can be:

```
export NCURSES_PATCHED_LIB="custom/path/to/patched/lib/files"
if [ -z "$LIBRARY_PATH" ]
then
    export LIBRARY_PATH="$NCURSES_PATCHED_LIB"
else
    LIBRARY_PATH="$LIBRARY_PATH:$NCURSES_PATCHED_LIB"
fi
```

Are you OK with this approach?

The patched NCURSES can be anywhere but it is developer's responsibility to properly setup NCURSES_PATCHED_LIB custom variable.

#17 - 05/02/2023 10:53 AM - Greg Shah

I am OK with this.

#18 - 05/02/2023 11:16 AM - Eugenie Lyzenko

Greg Shah wrote:

I am OK with this.

The next steps:

1. I'm creating 5167a branch to commit FWD changes.
2. While the 5167a is in progress I'm going to write new section in [Patching NCURSES](#) to completely describe the new libp2j.so building approach.
3. Merge 5167a into trunk explaining since this trunk version the FWD will use statically linked NCURSES calls to avoid any confusions with using the updated trunk.

Is this plan good?

#19 - 05/02/2023 11:36 AM - Roger Borrello

Are there steps to migrate from *dynamic-linked and patched* to *static linked/no-patch*? Once we have a trunk that includes the statically linked **ncurses**, what do we do to migrate? Perhaps I could put together a quick script to do so? Then the methods by which we build systems would need to switch over at that time, too.

#20 - 05/02/2023 11:44 AM - Eugenie Lyzenko

Roger Borrello wrote:

Are there steps to migrate from *dynamic-linked and patched* to *static linked/no-patch*? Once we have a trunk that includes the statically linked **ncurses**, what do we do to migrate?

The migration is to:

1. Set up and build new patched NCURSES with ability to access the NCURSES lib directory containing modified *.a files.
2. Properly set up LIBRARY_PATH system variable to be pointed to directory containing *.a files from NCURSES.
3. Full rebuild of FWD to use new approach.

Perhaps I could put together a quick script to do so? Then the methods by which we build systems would need to switch over at that time, too.

I'm not sure. The ready to use *.a object libraries from NCURSES should be ready before new FWD build can be started.

#21 - 05/02/2023 12:19 PM - Eugenie Lyzenko

I have updated instructions to use static linking. Please review. It is important everyone see it as clear and able to be done.

#22 - 05/02/2023 12:46 PM - Greg Shah

I will review the instructions shortly. We do need to make sure that we have a plan for all customers to migrate their FWD build process. That can be a step 2.5 in your plan.

#23 - 05/02/2023 12:58 PM - Eugenie Lyzenko

Greg Shah wrote:

I will review the instructions shortly. We do need to make sure that we have a plan for all customers to migrate their FWD build process. That can be a step 2.5 in your plan.

OK. As soon as we merge this into trunk I'll make proper update for replica system. Not clear what we can do more except making good instruction for NCURSES patches.

#24 - 05/02/2023 01:09 PM - Eugenie Lyzenko

The 5167a updated for review to revision 14554.

All required FWD changes to move to the NCURSES static linking.

#25 - 05/02/2023 02:27 PM - Greg Shah

Eugenie Lyzenko wrote:

I have updated instructions to use static linking. Please review. It is important everyone see it as clear and able to be done.

I think the document is very confusing. It was confusing before your edits. I think it is even more confusing after your edits since there are new sections talking about static linking. These are spread around in places and mixed in to text sections that have content for non-static cases.

We will need a single, clean document for static linking which has no references to the old system-wide patching process. Please start a fresh page for that.

We need to automate the process for creating the **local** (not system-wide) patched version. That is a script to do the patching. I assume that can be done by a non-sudo user, is that right?

We also should consider if we need to also have the apt hooks like before. I think it should be an option.

#26 - 05/02/2023 02:52 PM - Eugenie Lyzenko

Greg Shah wrote:

Eugenie Lyzenko wrote:

I have updated instructions to use static linking. Please review. It is important everyone see it as clear and able to be done.

I think the document is very confusing. It was confusing before your edits. I think it is even more confusing after your edits since there are new sections talking about static linking. These are spread around in places and mixed in to text sections that have content for non-static cases.

We will need a single, clean document for static linking which has no references to the old system-wide patching process. Please start a fresh page for that.

OK. In progress.

We need to automate the process for creating the **local** (not system-wide) patched version. That is a script to do the patching. I assume that can be done by a non-sudo user, is that right?

Yes, sudo is not required.

We also should consider if we need to also have the apt hooks like before. I think it should be an option.

OK.

#27 - 05/02/2023 03:52 PM - Eugenie Lyzenko

Added new page for [Patching NCURSES Using Static Linking](#) can be seen from [Building FWD From Source](#)

Please review.

#28 - 05/02/2023 06:41 PM - Eugenie Lyzenko

Greg Shah wrote:

We need to automate the process for creating the **local** (not system-wide) patched version. That is a script to do the patching. I assume that can be done by a non-sudo user, is that right?

Do we need this script to be the part of the FWD trunk repository? Or this should be a standalone unit can be downloaded separately?

#29 - 05/03/2023 11:52 AM - Greg Shah

Do we need this script to be the part of the FWD trunk repository? Or this should be a standalone unit can be downloaded separately?

No, this is separate, downloaded from the patching wiki page.

#30 - 05/03/2023 02:28 PM - Eugenie Lyzenko

- File *setup_ncurses6x.sh* added

Greg,

This is automated script for review. It should be run inside directory where the patched NCURSES to be installed. Also optionally adds system variable change to use generated object with FWD linkage.

#31 - 05/03/2023 03:56 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14554. New branch version is 14555.

#32 - 05/04/2023 08:32 AM - Roger Borrello

Should there be a location within the FWD directory structure to place the library for consistency? Like build/usrlib or similar? Then FWD build environments can have \$LIBRARY_PATH contain that path and not rely on a user's environment.

If we do need to update the user's environment, there is a ~/.gcdrc hook we set in the GCD Standard Desktop ~/.bashrc that might be a better place for updating LIBRARY_PATH.

I found that the same chunk is added multiple times to the ~/.bashrc file, whether it is already there or not. But again... if my first point is useful, we don't need to update the user's environment.

#33 - 05/04/2023 08:43 AM - Greg Shah

Should there be a location within the FWD directory structure to place the library for consistency? Like build/usrlib or similar? Then FWD build environments can have \$LIBRARY_PATH contain that path and not rely on a user's environment.

This is nice from the perspective that it reduces the amount of configuration. It is less nice if you have to build more than one version of FWD (i.e. multiple branches).

If we do need to update the user's environment, there is a ~/.gcdrc hook we set in the GCD Standard Desktop ~/.bashrc that might be a better place for updating LIBRARY_PATH.

I found that the same chunk is added multiple times to the ~/.bashrc file, whether it is already there or not. But again... if my first point is useful, we don't need to update the user's environment.

This is something to discuss in #6894, not here.

#34 - 05/04/2023 09:12 AM - Eugenie Lyzenko

Greg Shah wrote:

Should there be a location within the FWD directory structure to place the library for consistency? Like build/usrlib or similar? Then FWD build environments can have \$LIBRARY_PATH contain that path and not rely on a user's environment.

This is nice from the perspective that it reduces the amount of configuration. It is less nice if you have to build more than one version of FWD (i.e. multiple branches).

Currently we can install patched NCURSES in any arbitrary location. This a kind of freedom I guess. No need to attach to any projects or FWD instance.

If we do need to update the user's environment, there is a ~/.gcdrc hook we set in the GCD Standard Desktop ~/.bashrc that might be a better place for updating LIBRARY_PATH.

I found that the same chunk is added multiple times to the ~/.bashrc file, whether it is already there or not. But again... if my first point is useful, we don't need to update the user's environment.

Roger,

You have multiple chunks because the new console session should be opened to apply the environment change. I made special echo warning note

for this inside script.

#36 - 05/04/2023 09:57 AM - Roger Borrello

Eugenie Lyzenko wrote:

Roger,

You have multiple chunks because the new console session should be opened to apply the environment change. I made special echo warning note for this inside script.

Actually it was because my `/etc/apt/sources.list` had all the `deb-src` commented out, so when I ran the script it errored at the patching portion and added the updates to the `.bashrc` anyway.

#37 - 05/04/2023 11:00 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14555. New branch version is 14556.

#38 - 05/04/2023 11:31 AM - Eugenie Lyzenko

Roger Borrello wrote:

Eugenie Lyzenko wrote:

Roger,

You have multiple chunks because the new console session should be opened to apply the environment change. I made special echo warning note for this inside script.

Actually it was because my `/etc/apt/sources.list` had all the `deb-src` commented out, so when I ran the script it errored at the patching portion and added the updates to the `.bashrc` anyway.

I have updated the [Patching NCURSES Using Static Linking](#) section to include required actions to prepare software sources.

#39 - 05/04/2023 03:21 PM - Eugenie Lyzenko

- File deleted (setup_ncurses6x.sh)

#40 - 05/04/2023 03:22 PM - Eugenie Lyzenko

- File setup_ncurses6x.sh added

Roger Borrello wrote:

Eugenie Lyzenko wrote:

Roger,

You have multiple chunks because the new console session should be opened to apply the environment change. I made special echo warning note for this inside script.

Actually it was because my /etc/apt/sources.list had all the deb-src commented out, so when I ran the script it errored at the patching portion and added the updates to the .bashrc anyway.

I have updated the script to avoid this issue.

#41 - 05/04/2023 05:39 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14559. New branch version is 14560.

#42 - 05/05/2023 05:08 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14560. New branch version is 14561.

#43 - 05/06/2023 08:04 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14561. New branch version is 14562.

#44 - 05/09/2023 07:55 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14563. New branch version is 14564.

#45 - 05/10/2023 02:20 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14564. New branch version is 14565.

#46 - 05/10/2023 04:02 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14565. New branch version is 14566.

#47 - 05/11/2023 07:57 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14566. New branch version is 14567.

#48 - 05/12/2023 11:14 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14567. New branch version is 14568.

#49 - 05/12/2023 12:08 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14568. New branch version is 14569.

#50 - 05/12/2023 02:21 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14570. New branch version is 14571.

#51 - 05/15/2023 02:30 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14571. New branch version is 14572.

#52 - 05/18/2023 12:53 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14572. New branch version is 14573.

#53 - 05/19/2023 12:42 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14573. New branch version is 14574.

#54 - 05/19/2023 05:02 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14574. New branch version is 14575.

#55 - 05/20/2023 03:21 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14575. New branch version is 14576.

#56 - 05/22/2023 12:54 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14579. New branch version is 14580.

#57 - 05/23/2023 09:40 AM - Roger Borrello

I noticed there is an update to ncurses for Ubuntu 22.04. Would the patched static library need to be rebuilt after something like this?

```
rfb@rfb:~/counteract$ apt list --upgradable
Listing... Done
dpkg-dev/jammy-updates, jammy-updates 1.21.1ubuntu2.2 all [upgradable from: 1.21.1ubuntu2.1]
dpkg/jammy-updates 1.21.1ubuntu2.2 amd64 [upgradable from: 1.21.1ubuntu2.1]
libdpkg-perl/jammy-updates, jammy-updates 1.21.1ubuntu2.2 all [upgradable from: 1.21.1ubuntu2.1]
libegl-mesa0/jammy-updates 22.2.5-0ubuntu0.1~22.04.2 amd64 [upgradable from: 22.2.5-0ubuntu0.1~22.04.1]
libgbml/jammy-updates 22.2.5-0ubuntu0.1~22.04.2 amd64 [upgradable from: 22.2.5-0ubuntu0.1~22.04.1]
libgl1-mesa-dri/jammy-updates 22.2.5-0ubuntu0.1~22.04.2 amd64 [upgradable from: 22.2.5-0ubuntu0.1~22.04.1]
libglapi-mesa/jammy-updates 22.2.5-0ubuntu0.1~22.04.2 amd64 [upgradable from: 22.2.5-0ubuntu0.1~22.04.1]
libglx-mesa0/jammy-updates 22.2.5-0ubuntu0.1~22.04.2 amd64 [upgradable from: 22.2.5-0ubuntu0.1~22.04.1]
libncurses-dev/jammy-security 6.3-2ubuntu0.1 amd64 [upgradable from: 6.3-2]
libncurses6/jammy-security 6.3-2ubuntu0.1 amd64 [upgradable from: 6.3-2]
libncursesw6/jammy-security 6.3-2ubuntu0.1 amd64 [upgradable from: 6.3-2]
libtinfo-dev/jammy-security 6.3-2ubuntu0.1 amd64 [upgradable from: 6.3-2]
...
```

#58 - 05/23/2023 09:41 AM - Greg Shah

Need to be? No.

Should be? Yes.

#59 - 05/23/2023 10:08 AM - Eugenie Lyzenko

My understanding is we want to be completely independent from system wide ncurses. Another point is to ensure system backward compatibility for 18.04 or even less of the Ubuntu. This can mean we need to use the version that is fully supported by most earlier version of the OS that is planning to declare as compatible one. What if the static code from 22.04 will have some issues in 18.04 for example?

The 5167a rebased with trunk revision 14583. New branch version is 14584.

#60 - 05/23/2023 10:14 AM - Greg Shah

My understanding is we want to be completely independent from system wide ncurses.

Yes

Another point is to ensure system backward compatibility for 18.04 or even less of the Ubuntu. This can mean we need to use the version that is fully supported by most earlier version of the OS that is planning to declare as compatible one.

This is less important. We would want customers to be able to rebuild FWD on older versions, but we won't guarantee that our code will work without modifications. FWD v4 will be supported on 22.04 LTS and not on older versions.

#61 - 05/25/2023 10:07 PM - Eugenie Lyzenko

- File deleted (*setup_ncurses6x.sh*)

#62 - 05/25/2023 10:07 PM - Eugenie Lyzenko

- File *setup_ncurses6x.sh* added

The 5167a rebased with trunk revision 14586. Updated to revision 14588.

This update adds preferred usage of the include files from custom NCURSES patched static location. Also I have attached the modified script to handle this usage properly. Just tested with fresh Ubuntu 22.04 install. Works fine. So this can be considered as unified for Ubuntu 18.04+ versions. So the 6.x patches can be the only one we need to build. No need to keep different versions of the patches for old 5.x versions.

A bit update to Wiki for install instruction. We need to ensure dpkg-dev exists on developer system.

So my suggest is to merge this into trunk and start using static libp2j.so. This can give us some degree of freedom from system side NCURSES updates (I found last week NCURSES was updated again).

What do you think? Are we ready to make this step?

#63 - 05/26/2023 08:22 AM - Greg Shah

- Status changed from WIP to Review

Roger: Please review.

Eugenie: What do we need to do to migrate each project? I want to make sure we don't create breakage or chaos when this is merged to trunk.

#64 - 05/26/2023 10:12 AM - Eugenie Lyzenko

Greg Shah wrote:

Eugenie: What do we need to do to migrate each project? I want to make sure we don't create breakage or chaos when this is merged to trunk.

1. Execute script setup_ncurses6x.sh only once on every development system and check if it was completed without issues.
2. Rebuild and re-deploy FWD for all related projects.
3. The developer should know what is changed and how it works now. This is important. For example the directory with static NCURSES should permanently exist for FWD compilation to be OK with proper libraries.

#65 - 05/26/2023 10:22 AM - Greg Shah

Rebuild and re-deploy FWD for all related projects.

This will require changes to the current processes used for each project and possibly also will require scripting changes for anything automated. We need to document this in detail.

#66 - 05/26/2023 10:33 AM - Eugenie Lyzenko

Greg Shah wrote:

Rebuild and re-deploy FWD for all related projects.

This will require changes to the current processes used for each project and possibly also will require scripting changes for anything automated...

Please clarify. I thought only FWD will need to be replaced. The preparation step should be done only once in separate terminal session. As soon as the new NCURSES build is ready to use - the rest is as it was before cycle (update FWD, rebuild FWD).

#67 - 05/26/2023 10:36 AM - Greg Shah

Don't we need to configure FWD to know about where the static library is located as part of the FWD build?

#68 - 05/26/2023 10:41 AM - Eugenie Lyzenko

Greg Shah wrote:

Don't we need to configure FWD to know about where the static library is located as part of the FWD build?

No. It will be ready for every new console session started after set up script is completed (or just re-login Ubuntu).

#69 - 05/26/2023 10:56 AM - Greg Shah

Does the setup script replace the system-wide ncurses? If not, then how does the FWD build know where the libraries are?

#70 - 05/26/2023 11:02 AM - Eugenie Lyzenko

Greg Shah wrote:

Does the setup script replace the system-wide ncurses?

No, the system wide NCURSES package will be untouched.

If not, then how does the FWD build know where the libraries are?

Via LIBRARY_PATH change the script does. Making patched libraries in a head of the library list to link with static version.

#71 - 05/26/2023 12:58 PM - Greg Shah

If not, then how does the FWD build know where the libraries are?

Via LIBRARY_PATH change the script does. Making patched libraries in a head of the library list to link with static version.

This means every build of FWD needs LIBRARY_PATH to be configured in some way. We have to provide some guidance on this for the migration.

#72 - 05/26/2023 04:50 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14592. Updated to revision 14594.

#73 - 05/26/2023 04:56 PM - Eugenie Lyzenko

Greg Shah wrote:

If not, then how does the FWD build know where the libraries are?

Via LIBRARY_PATH change the script does. Making patched libraries in a head of the library list to link with static version.

This means every build of FWD needs LIBRARY_PATH to be configured in some way. We have to provide some guidance on this for the migration.

Do you think we need to keep the co-existence of the old way built (dynamic NCURSES) and new static libp2j.so simultaneously?

Certainly the new approach background must be clearly documented. On the other hand on the fresh install the LIBRARY_PATH is empty. If not the script does not replace with NCURSES lib path. Instead the NCURSES libs are becoming in a head of the current list.

#74 - 05/26/2023 05:01 PM - Greg Shah

Do you think we need to keep the co-existence of the old way built (dynamic NCURSES) and new static libp2j.so simultaneously?

I think it would be the safe thing to do. I prefer not to force everyone to migrate immediately. If using this new version defaults to the current approach that is good. When someone does migrate, the code should pick that up and honor it.

#75 - 05/26/2023 05:58 PM - Eugenie Lyzenko

Greg Shah wrote:

Do you think we need to keep the co-existence of the old way built (dynamic NCURSES) and new static libp2j.so simultaneously?

I think it would be the safe thing to do. I prefer not to force everyone to migrate immediately. If using this new version defaults to the current approach that is good. When someone does migrate, the code should pick that up and honor it.

How about this:

1. We introduce new environment variable, say NCURSES_PATCHED pointing to the give location of the custom static libraries root. For current libp2j.so it is undefined.
2. The makefile for libp2j.so will check if the NCURSES_PATCHED exist, if no the current version is producing, if yes - we will have static version.
3. The setup_ncurses6x.sh script adds this variable and builds new libraries to link. After execution this script only new static libp2j.so will be produced for next FWD building.
4. If for some reasons it is required to get back to current version the NCURSES_PATCHED environment can be manually disabled again and next FWD building will produce dynamic libraries. No need to remove previously compiled static NCURSES libraries. They will never used if NCURSES_PATCHED disabled.
5. We can even avoid to modify LIBRARY_PATH for static building with this approach to not interfere with other C projects compilations.
6. The only restriction is at any given time only one libp2j.so be built, either current dynamic or new static.

What do you think? Is it OK?

#76 - 05/28/2023 11:48 PM - Roger Borrello

Eugenie Lyzenko wrote:

How about this:

1. We introduce new environment variable, say NCURSES_PATCHED pointing to the give location of the custom static libraries root.

Should it be NCURSES_STATIC?

#77 - 05/29/2023 07:45 AM - Eugenie Lyzenko

Roger Borrello wrote:

Eugenie Lyzenko wrote:

How about this:

1. We introduce new environment variable, say NCURSES_PATCHED pointing to the give location of the custom static libraries root.

Should it be NCURSES_STATIC?

Yes this has better explanation/hint for what we have there. Or may be NCURSES_PATCHED_STATIC? Or NCURSES_FWD_STATIC?

#78 - 05/29/2023 08:54 AM - Greg Shah

Nice idea guys, NCURSES_FWD_STATIC is a good choice.

#79 - 05/29/2023 08:56 AM - Greg Shah

What do you think? Is it OK?

Yes, this is a good plan.

#80 - 05/29/2023 11:59 AM - Eugenie Lyzenko

- File deleted (*setup_ncurses6x.sh*)

#81 - 05/29/2023 12:00 PM - Eugenie Lyzenko

- File *setup_ncurses6x.sh* added

The 5167a updated for review to revision 14595.

Reworked general approach. Used NCURSES_FWD_STATIC to separate old and new linkage types. Now compatible with current approach. The LIBRARY_PATH is not used anymore to make static version.

The updated script is updated accordingly.

#82 - 05/31/2023 04:16 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14594. The new revision is 14597.

#83 - 06/01/2023 09:10 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14599. The new revision is 14602.

#84 - 06/02/2023 08:16 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14605. The new revision is 14608.

#85 - 06/03/2023 05:23 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14606. The new revision is 14609.

#86 - 06/05/2023 04:25 PM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14612. The new revision is 14615.

#87 - 06/06/2023 09:32 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14614. The new revision is 14617.

#88 - 06/06/2023 09:54 AM - Greg Shah

Roger: Please test 5167a with a real customer application using both static and non-static approaches.

#89 - 06/06/2023 09:57 AM - Roger Borrello

Greg Shah wrote:

Roger: Please test 5167a with a real customer application using both static and non-static approaches.

Is it acceptable to work with an application that has been converted with **trunk-14592**?

#90 - 06/06/2023 10:05 AM - Eugenie Lyzenko

The 5167a rebased with trunk revision 14615. The new revision is 14618.

#91 - 06/06/2023 10:07 AM - Eugenie Lyzenko

Roger Borrello wrote:

Greg Shah wrote:

Roger: Please test 5167a with a real customer application using both static and non-static approaches.

Is it acceptable to work with an application that has been converted with **trunk-14592**?

Yes, it is pretty independent from rest of the FWD. No need to re-convert anything. You can use only one new makefile from 5167a. And static ncurses setup script.

#92 - 06/06/2023 10:13 AM - Roger Borrello

Is the testing methodology to build FWD with/without:

```
export NCURSES_FWD_STATIC=/home/rfb/projects/fwd/ncurses/ncurses-6.3
```

in my environment, and utilize the resultant libp2j.so in the application's configuration for verification?

Does it make sense to build both versions with different names, and the application deployment would need to be smart enough to choose one over the default?

#93 - 06/06/2023 10:22 AM - Eugenie Lyzenko

Roger Borrello wrote:

Is the testing methodology to build FWD with/without:
[...]
in my environment, and utilize the resultant libp2j.so in the application's configuration for verification?

Yes, defining NCURSES_FWD_STATIC turns on the static version, removing NCURSES_FWD_STATIC from environment will back to currently used dynamic one.

I think no need to have both libraries with different names. You can identify the static version as bigger in size. Also our code depends on libp2j.so exact name.

#94 - 06/06/2023 10:46 AM - Roger Borrello

Eugenie Lyzenko wrote:

I think no need to have both libraries with different names. You can identify the static version as bigger in size. Also our code depends on libp2j.so exact name.

Bigger can only be determined if you see both versions at the same time. I'm looking at:

```
rfb@rfb:~/projects/fwd/5167a$ find ./build/ -name "*.so" -exec ls -l {} \;  
-rwxrwx--- 1 rfb rfb 66320 Jun  6 10:01 ./build/native/libp2j.so  
-rw-rw---- 1 rfb rfb 66320 Jun  6 10:01 ./build/lib/libp2j.so
```

So I cannot tell if it is static or dynamic. I would propose libp2j_s.so and libp2j_d.so and libp2j.so would be the dynamic one. My application deployment would then need to put the static one in place, if that's the one I wanted to use.

#95 - 06/06/2023 11:03 AM - Eugenie Lyzenko

Roger Borrello wrote:

Eugenie Lyzenko wrote:

I think no need to have both libraries with different names. You can identify the static version as bigger in size. Also our code depends on libp2j.so exact name.

Bigger can only be determined if you see both versions at the same time. I'm looking at:
[...]

It is the dynamic ones. The static has 269920 bytes in size.

So I cannot tell if it is static or dynamic. I would propose libp2j_s.so and libp2j_d.so and libp2j.so would be the dynamic one. My application deployment would then need to put the static one in place, if that's the one I wanted to use.

Do we really need this complication? The current limitation is you can build either static or dynamic, not both. If we consider this change as improvement why do we need to keep both versions working on the same machine? If the new version is not an improvement - why do we need the static version? The ability to have both building is a chance for developer to postpone migration to new code to convenient time instead of forcing do this mandatory right now.

Am I missing for something?

#96 - 06/06/2023 11:34 AM - Roger Borrello

Greg Shah wrote:

Roger: Please test 5167a with a real customer application using both static and non-static approaches.

I tried ChUI with both, and I couldn't tell the difference. I even noticed the same oddities with trying to maneuver through a field with the arrow keys. The keystrokes didn't seem to be honored. But since this happened whether I was using this branch or 4938a, it's not for this task.

As for whether both flavors make sense, if we are planning that everyone would abandon the arduous ncurses patching, I don't see the need.

#97 - 06/06/2023 11:51 AM - Greg Shah

Yes, let's avoid the more complex naming. If everything is working, then we can move ahead with this branch.

#98 - 06/06/2023 01:03 PM - Greg Shah

if we are planning that everyone would abandon the arduous ncurses patching

To be clear, this doesn't avoid patching. It avoids patching the system version of ncurses.

#99 - 06/06/2023 01:18 PM - Roger Borrello

Greg Shah wrote:

if we are planning that everyone would abandon the arduous ncurses patching

To be clear, this doesn't avoid patching. It avoids patching the system version of ncurses.

Got it. The environment that is a recipient of runtime support only doesn't require the ncurses package at all, correct?

#100 - 06/06/2023 01:25 PM - Eugenie Lyzenko

Roger Borrello wrote:

Greg Shah wrote:

if we are planning that everyone would abandon the arduous ncurses patching

To be clear, this doesn't avoid patching. It avoids patching the system version of ncurses.

Got it. The environment that is a recipient of runtime support only doesn't require the ncurses package at all, correct?

Yes, all ncurses calls used by FWD will be inside libp2j.so static code.

#101 - 06/06/2023 01:39 PM - Roger Borrello

The container project to build FWD Docker images will need to modify the "patching" process.

Currently it unzips the patch process to the base image, and performs the manage_term_patches.sh to setup the environment, which will also have the hook for apt.

Archive: fwd_ubuntu/fix_ncurses.zip			
Length	Date	Time	Name
-----	-----	-----	----
769	2022-05-20	16:56	root/manage_term_patches.sh
1846	2022-05-20	16:10	root/patch_ncurses6.sh
585	2022-05-19	17:28	root/patch_terminfo.sh
561	2022-05-19	17:28	root/ncurses_curses_h_in_v6.1_20200708.patch
1044	2022-05-19	17:28	root/ncurses_lib_getch_c_v6.1_20200708.patch
64	2022-05-19	17:31	etc/apt/apt.conf.d/95patch-ncurses
-----			-----
4869			6 files

Will it be enough to just perform the setup_ncurses6x.sh so that a patched library and the NCURSES_FWD_STATIC environment variable is setup?
Or is there an ongoing need to hook apt still?

#102 - 06/06/2023 02:04 PM - Eugenie Lyzenko

Roger Borrello wrote:

The container project to build FWD Docker images will need to modify the "patching" process.

Currently it unzips the patch process to the base image, and performs the manage_term_patches.sh to setup the environment, which will also have the hook for apt.
[...]

Will it be enough to just perform the setup_ncurses6x.sh so that a patched library and the NCURSES_FWD_STATIC environment variable is setup?

Yes, it should be enough to run this once per system. The idea was to make a life as simple as possible here.

Or is there an ongoing need to hook apt still?

I do not know any such requirements.

#103 - 06/06/2023 02:23 PM - Roger Borrello

Eugenie Lyzenko wrote:

Or is there an ongoing need to hook apt still?

I do not know any such requirements.

I believe that hook is there because there is a chance that an apt install that occurs could pull in an unpatched ncurses library. So if we are statically linked, we won't care about that anymore. I would think we'd only really have to perform the patch process again if we are linking to updated ncurses. :-)

#104 - 06/06/2023 03:19 PM - Eugenie Lyzenko

Roger Borrello wrote:

Eugenie Lyzenko wrote:

Or is there an ongoing need to hook apt still?

I do not know any such requirements.

I believe that hook is there because there is a chance that an apt install that occurs could pull in an unpatched ncurses library. So if we are statically linked, we won't care about that anymore.

Yes.

I would think we'd only really have to perform the patch process again if we are linking to updated ncurses. :-)

Do you mean when new system wide ncurses code has changed our patched static version is also to be updated? We can update our patched code or not. This should not affect the FWD. It will always use working patched ncurses. Up to date or not - not important. The developer can update it in spare time. Our static lib is completely independent on system ncurses (which always exists on Linux on some runtime level). The makefile in FWD uses patched library first to link, no matter what is the state of system libraries.

#105 - 06/06/2023 05:29 PM - Greg Shah

I think we just create a new patched ncurses every time we build FWD for Docker. It will always be the latest available for that build. Then it doesn't change over the lifetime of the Docker image.

#106 - 06/06/2023 05:39 PM - Greg Shah

- % Done changed from 90 to 100

You can go ahead and merge to trunk. Please provide documentation on how to use it in the merge email.

#107 - 06/06/2023 05:46 PM - Eugenie Lyzenko

Greg Shah wrote:

You can go ahead and merge to trunk. Please provide documentation on how to use it in the merge email.

I need to upload/copy setup_ncurses6x.sh to the Redmine download area. Can you provide instruction for how to do this or point me where it is documented?

#108 - 06/06/2023 06:36 PM - Eugenie Lyzenko

Branch 5167a was merged to trunk as revno 14616 then it was archived.

This change will be on replica later today if no objections.

#109 - 06/07/2023 10:16 AM - Greg Shah

- Status changed from Review to Closed

I've renamed and made edits to the wiki page ([Patching NCURSES Using Static Linking](#)). I've also linked it into some other locations.

Eugenie: Please review that page. Let me know if I got anything wrong.

#110 - 06/07/2023 12:38 PM - Eugenie Lyzenko

Greg Shah wrote:

I've renamed and made edits to the wiki page ([Patching NCURSES Using Static Linking](#)). I've also linked it into some other locations.

Eugenie: Please review that page. Let me know if I got anything wrong.

I made some minor changes to sync the manual instructions with the recent set up script. The rest is OK. Please review.

#111 - 06/07/2023 12:51 PM - Roger Borrello

Greg Shah wrote:

I think we just create a new patched ncurses every time we build FWD for Docker. It will always be the latest available for that build. Then it doesn't change over the lifetime of the Docker image.

For Docker, I am placing the setup_ncurses6x.sh script in /tmp and run it so that the files are stored in /opt/ncurses. That's what I want NCURSES_FWD_STATIC to point to. The script will be removed. To accomplish this, I just need to run the script from the /opt/ncurses directory, correct?

I also plan to update the /etc/environment with NCURSES_FWD_STATIC=/opt/ncurses instead of the .bashrc so that the variable is available across any signon, and not associated with a specific user. To do this, I will need to make updates to the script so that it can receive a specific directory, instead of using pwd.

Then there's patching terminfo. Is that now a one-time occurrence, or does that need to stay in the manage_term_patches.sh setup, along with the apt exit?

#112 - 06/07/2023 02:03 PM - Eugenie Lyzenko

Roger Borrello wrote:

Greg Shah wrote:

I think we just create a new patched ncurses every time we build FWD for Docker. It will always be the latest available for that build. Then it doesn't change over the lifetime of the Docker image.

For Docker, I am placing the setup_ncurses6x.sh script in /tmp and run it so that the files are stored in /opt/ncurses. That's what I want NCURSES_FWD_STATIC to point to. The script will be removed. To accomplish this, I just need to run the script from the /opt/ncurses directory, correct?

I also plan to update the /etc/environment with NCURSES_FWD_STATIC=/opt/ncurses instead of the .bashrc so that the variable is available across any signon, and not associated with a specific user. To do this, I will need to make updates to the script so that it can receive a specific directory, instead of using pwd.

Then there's patching terminfo. Is that now a one-time occurrence, or does that need to stay in the manage_term_patches.sh setup, along with the apt exit?

I think terminfo is still need to be patched. It is not directly dependent on NCURSES. So for terminfo now nothing has changed with moving to statically linked libp2j.so. I guess patch_terminfo.sh should be enough. No need to do all from manage_term_patches.sh.

#113 - 06/07/2023 03:26 PM - Roger Borrello

Thanks... one more docker-related question. The docker image contained some runtime support packages:

- libncurses5-dev
- ncurses-term

Are these still required to be loaded via apt-get install? They were amongst these others: ARG RUNTIME_TOOLS="dpkg-dev libffi-dev libncurses5-dev ncurses-term ant libpam0g-dev"

#114 - 06/07/2023 05:59 PM - Eugenie Lyzenko

Roger Borrello wrote:

Thanks... one more docker-related question. The docker image contained some runtime support packages:

- libncurses5-dev
- ncurses-term

Are these still required to be loaded via apt-get install? They were amongst these others: ARG RUNTIME_TOOLS="dpkg-dev libffi-dev libncurses5-dev ncurses-term ant libpam0g-dev"

Thanks for this question. I have updated the static ncurses instruction section.

Both packages are **not** needed if **only** statically linked libp2j.so is planned to be used. I have tested on Ubuntu 22.04. Removed libncurses-dev, ncurses-term was not even installed. The FWD building is OK with statically linked native library is such conditions.

#115 - 06/07/2023 06:31 PM - Roger Borrello

- File *setup_ncurses6x.sh* added

Eugenie Lyzenko wrote:

Thanks for this question. I have updated the static ncurses instruction section.

Both packages are **not** needed if **only** statically linked libp2j.so is planned to be used. I have tested on Ubuntu 22.04. Removed libncurses-dev, ncurses-term was not even installed. The FWD building is OK with statically linked native library is such conditions.

Great... I am thinning down the Docker images thanks to this task, since we are only utilizing statically linked ncurses.

I am attaching the updated setup_ncurses6x.sh that functions within the Docker container for review. It should operate the same on a normal system without any parameters:

Usage: `./setup_ncurses6x.sh [-p<path>] [-sd]`

Where:

```
p    = Use specific path instead of current directory
s    = Set system-wide environment instead of just current user
d    = Dry run. Just show commands.
```

The -p option lets you position only the lib and include directories into the path specified. The NCURSE_FWD_STATIC environment variable is then set to that location. The -s allows you to specify across all users. The -d is self-explanatory.

#116 - 06/07/2023 07:07 PM - Roger Borrello

- billable changed from No to Yes
- vendor_id deleted (GCD)
- File setup_ncurses6x.sh added
- Assignee changed from Eugenie Lyzenko to Eric Faulhaber
- Priority changed from Normal to Low

Actually this one attached I have made updates that should work on Red Hat. I based it off the patch_ncurses6-centos.sh we had used on a customer environment. I will also need to update the gcd_systems project to support the static linking on our workstations.

#117 - 06/07/2023 07:09 PM - Eugenie Lyzenko

Roger Borrello wrote:

Eugenie Lyzenko wrote:

Thanks for this question. I have updated the static ncurses instruction section.

Both packages are **not** needed if **only** statically linked libp2j.so is planned to be used. I have tested on Ubuntu 22.04. Removed libncurses-dev, ncurses-term was not even installed. The FWD building is OK with statically linked native library is such conditions.

Great... I am thinning down the Docker images thanks to this task, since we are only utilizing statically linked ncurses.

I am attaching the updated setup_ncurses6x.sh that functions within the Docker container for review. It should operate the same on a normal system without any parameters:

[...]

The -p option lets you position only the lib and include directories into the path specified. The NCURSE_FWD_STATIC environment variable is then set to that location. The -s allows you to specify across all users. The -d is self-explanatory.

Roger,

This new script has at least 2 issues:

1. It does not remove previous ncurses* subdirectory, so patching become problematic.
2. It makes the terminal message area messed.

```
...
++'
+ cd include
+ ln -s curses.h ncurses.h
+ cd ..
+ echo +-----+
+-----+
+ echo '|                Patched NCURSES libraries update completed.                |'
|                Patched NCURSES libraries update completed.                |
+ [[ ! -z '' ]]
+ '[' -z /home/evl/build_factory/ncurses_static/ncurses-6.1 ']'
```

```
+ echo '|           Please do FULL FWD rebuild to activate the changes.'|'|
|           Please do FULL FWD rebuild to activate the changes.      |
+ echo +-----+
+-----+
+ popd
...
```

Can you please resolve it. Or make another script to be run inside docker instead of using same name.

#118 - 06/07/2023 07:12 PM - Roger Borrello

- *billable changed from Yes to No*
- *Assignee changed from Eric Faulhaber to Eugenie Lyzenko*
- *Priority changed from Low to Normal*
- *vendor_id set to GCD*

So sorry... I had posted the version with "cr*p" in it. The subsequent one that supports Red Hat should work better.

Redmine messed up the header, but it looks like you corrected it.

#119 - 06/07/2023 07:30 PM - Eugenie Lyzenko

- *vendor_id deleted (GCD)*
- *Priority changed from Normal to Low*
- *billable changed from No to Yes*

Roger Borrello wrote:

So sorry... I had posted the version with "cr*p" in it. The subsequent one that supports Red Hat should work better.

OK. The last one works fine. I have updated the NCURSES setup document page with adding possible options for your new script.

Please remove the previous problematic script file to ensure the correct one will be used.

Greg,

Seems like we need to update Redmine download directory replacing setup_ncurses6x.sh with recent one from Roger.

#120 - 06/07/2023 07:31 PM - Eugenie Lyzenko

- Priority changed from Low to Normal

#121 - 06/07/2023 09:10 PM - Roger Borrello

- File deleted (*setup_ncurses6x.sh*)

#122 - 06/07/2023 09:18 PM - Roger Borrello

Eugenie Lyzenko wrote:

Please remove the previous problematic script file to ensure the correct one will be used.

Done.

#123 - 06/08/2023 04:54 AM - Greg Shah

Eugenie Lyzenko wrote:

Greg Shah wrote:

I've renamed and made edits to the wiki page ([Patching NCURSES Using Static Linking](#)). I've also linked it into some other locations.

Eugenie: Please review that page. Let me know if I got anything wrong.

I made some minor changes to sync the manual instructions with the recent set up script. The rest is OK. Please review.

Don't we need libncurses-dev to be installed?

Also, the version of setupncurses6x.sh that is attached to the wiki page doesn't have any command line parameters.

#124 - 06/08/2023 04:57 AM - Greg Shah

Then there's patching terminfo. Is that now a one-time occurrence, or does that need to stay in the `manage_term_patches.sh` setup, along with the apt exit?

I think terminfo is still need to be patched. It is not directly dependent on NCURSES. So for terminfo now nothing has changed with moving to statically linked libp2j.so. I guess patch_terminfo.sh should be enough. No need to do all from manage_term_patches.sh.

Correct. We do have [#5568](#) to see if we can move to an application-specific / private TERMINFO database. Until we work that task, patching it is needed.

#125 - 06/08/2023 05:02 AM - Greg Shah

Seems like we need to update Redmine download directory replacing setup_ncurses6x.sh with recent one from Roger.

Done. I think you have the rights to upload a new one even if you can't delete the old one. I did that now. The link automatically picks up the latest version of the file with that same name.

#126 - 06/08/2023 07:39 AM - Eugenie Lyzenko

Greg Shah wrote:

Eugenie Lyzenko wrote:

Greg Shah wrote:

I've renamed and made edits to the wiki page ([Patching NCURSES Using Static Linking](#)). I've also linked it into some other locations.

Eugenie: Please review that page. Let me know if I got anything wrong.

I made some minor changes to sync the manual instructions with the recent set up script. The rest is OK. Please review.

Don't we need libncurses-dev to be installed?

No. For static link approach the required include files are in patched custom directory. The libncurses-dev is required for dynamic linking compatibility only.

#127 - 07/21/2023 08:58 AM - Roger Borrello

I have made an update to the setup_ncurses6x.sh that is part of the gcd_systems project. It no longer updates the /etc/environment file with the NCURSES_FWD_STATIC environment variable if you request it to be system-wide, but rather creates /etc/profile.d/10-ncurses.sh to set the variable. This is more in line with best practices for system-wide environment variables.

A question to be able to respond to Tomasz's question posed in #6214-806:

Why not support both configurations - shared and static ? As FWD now requires an explicit setup to use static NCURSES (with a dedicated ENV variable) and I guess there are some setups still in shared mode which will not work in a current container.

Isn't the desire to sunset the shared library?

#128 - 07/21/2023 09:00 AM - Greg Shah

Isn't the desire to sunset the shared library?

Yes

#130 - 12/06/2023 06:41 PM - Roger Borrello

I fixed an oversight in files/setup_ncurses6x.sh such that the environment wasn't being updated properly, as well as it would continue to append the same wrong statement into the file:

```
--- /tmp/brz-diff-dzn531u_/old/roles/patch_ncurses/files/setup_ncurses6x.sh
+++ /home/rfb/projects/gcd_systems_master/roles/patch_ncurses/files/setup_ncurses6x.sh
@@ -100,7 +100,8 @@
     if [[ "$system" == "true" ]]; then
         rc_filename='/etc/profile.d/10-ncurses.sh'
         msg="|          The environment has been changed! Reboot to activate changes          |"
-        update="NCURSES_FWD_STATIC=\"\$PWD\" "
+        update="export NCURSES_FWD_STATIC=\"\$PWD\" "
+        rm $rc_filename
     else
         rc_filename=$HOME'/.bashrc'
         msg="|          The .bashrc has been changed! Please open new console to activate setup!          |"
```

To get this fix requires a pull of the gcd_system project, and running:

```
ansible-playbook -i hosts -K standard_workstation.yml
```

Or the local `/etc/profile.d/10-ncurses.sh` can be updated by hand.

#131 - 12/07/2023 03:03 PM - Greg Shah

Is this something for the standard workstation or is it a FWD issue? You are posting in a task about FWD, but your changes seem related to the standard workstation.

#132 - 12/07/2023 03:56 PM - Roger Borrello

Greg Shah wrote:

Is this something for the standard workstation or is it a FWD issue? You are posting in a task about FWD, but your changes seem related to the standard workstation.

It is kind of both. The setup of the environment across the system wasn't occurring properly. *And* it should be something we have everybody know about via a standard workstation update.

#133 - 12/07/2023 04:03 PM - Greg Shah

Is there some change in the scripts that our customers must use? If so, then it is a FWD issue.

If this is just something internal to GCD, then it is standard workstation and not something we care about here.

#134 - 12/07/2023 06:13 PM - Roger Borrello

Greg Shah wrote:

Is there some change in the scripts that our customers must use? If so, then it is a FWD issue.

It is `setup_ncurses6x.sh`. We make that available to customers via https://proj.goldencode.com/attachments/download/18526/setup_ncurses6x.sh to setup their development environment.

#135 - 12/08/2023 07:53 AM - Greg Shah

Then how do we provide a newer version outside of the standard workstation?

#136 - 12/12/2023 05:18 PM - Roger Borrello

Greg Shah wrote:

Then how do we provide a newer version outside of the standard workstation?

I'm not sure. Is the link off the wiki appropriate?

#137 - 12/12/2023 05:19 PM - Greg Shah

Yes

#138 - 12/12/2023 05:35 PM - Roger Borrello

- File `setup_ncurses6x.sh` added

#139 - 12/12/2023 05:43 PM - Roger Borrello

Updated the [wiki](#). Should the older versions be removed? I don't have authority.

#140 - 12/13/2023 06:43 AM - Greg Shah

We can leave them.

Files

setup_ncurses6x.sh	2.01 KB	05/29/2023	Eugenie Lyzenko
setup_ncurses6x.sh	3.93 KB	06/07/2023	Roger Borrello
setup_ncurses6x.sh	4.02 KB	12/12/2023	Roger Borrello