

Overview

This guide will show you how to generate and package an RPM file for Sangoma Hardware Drivers installation. Normally, user is able to compile the Wanpipe drivers on a “development” system - a system with the necessary compilers and Linux sources.

The purpose of this guide is to enable users to generate an rpm installation package for Netborder Express Linux drivers. This would give the capacity to install those drivers on Linux systems that do not have the necessary compilers and sources to build the driver at install time.

To generate drivers RPM package, please follow these steps:

- Step 1, Ensure your system meets all requirements described in the [Tools and pre-Requisites](#) section.
- Step 2, Generate the drivers RPM package following one of the two procedures described in:
 - [Standard Drivers RPM package generation](#) section
 - and
 - [Custom Drivers RPM package generation](#) section.

For the **Standard package generation** (recommended procedure), the resulting **.rpm** package will allow user to install Netborder Express hardware drivers on a Linux system that has the same kernels as the ones specified while generating the package.

For the **Custom package generation**, the resulting **.rpm** package enables user to have custom built hardware drivers packaged into an RPM file that allows user install this custom package on a target system.

Tools and pre-Requisites

1. A Linux system that have the following Linux rpm packages installed and functional:

- *kernel-devel*
- *ncurses-devel*
- *libtermcap-devel*
- *bison*
- *libtool*
- *flex*
- *gcc-c++*
- *automake*
- *autoconf*
- *imake*
- *rpm-lib*
- *net-snmp*

All listed above rpm packages can be obtained by running the command “***yum install <package_name>***” on a linux Centos 5.x. System connected to the internet. The ***<package_name>*** tag has to be replaced with the name of the package to install.

Note that the linux system **MUST** also have ***/bin/sh*** installed.

2. A Pre-Installed **Linux full** version of **NetBorder Express Gateway** (**netborder-gateway** package)

Standard Drivers RPM package generation (recommended)

1. **cd** to the following directory **XXX_INTALLDIR_XXX/drivers/rpm** where **XXX_INTALLDIR_XXX** is Netborder Express Gateway installation directory (usually **/opt/Netborder/Express/Gateway**).

2. From that directory run the following command:

./generate-drivers-rpm-package.sh [optional command argument]

The **[optional command argument]** is not mandatory to successfully run this command. The only supported arguments for the Current Netborder Express version are:

- **--with-linux:** This argument can be used to specify the absolute path that points to a configured (.config file must exist) kernel source. If path is not supplied, this argument is defaulted to **/lib/modules/<kernel_version>/build** where **<kernel_version>** is obtained using the **uname -r** command.

Usage Example:

./generate-drivers-rpm-package.sh --with-linux=<absolute_kernel_source_path>

3. Once the command execution succeeds, an RPM package named **wanpipe-<kernel_version>-<wanpipe_version>.rpm** is generated under **/usr/src/redhat/RPMS/i386**.

4. To install Netborder Express Hardware drivers on a target system, just copy the **.rpm** package generated at step 3. Then, from the directory where you've copied that package, run the following command: **rpm -ivh --force <rpm name>**, where **<rpm name>** is the name of the package generated at step 3.

Note that the target system **MUST** have the same kernel as the one specified while generating the rpm package (**--with-linux** argument value if any provided or the same kernel version as the system on which the package was built).

Custom Drivers RPM package generation

1. **cd** to the following directory `XXX_INTALLDIR_XXX/drivers/wanpipe` where `XXX_INTALLDIR_XXX` is Netborder Express Gateway installation directory (usually `/opt/Netborder/Express/Gateway`).
2. Create a directory in which you will virtually install Netborder Express hardware drivers and define an environment variable named `SANGOMA_DRV_RPM_BUILD_ROOT` to the absolute path of that directory.

Example:

```
- cd XXX_INTALLDIR_XXX/drivers/wanpipe
- mkdir rpmbuild
- export SANGOMA_DRV_RPM_BUILD_ROOT=XXX_INTALLDIR_XXX/drivers/wanpipe/rpmbuild
```

3. From the `XXX_INTALLDIR_XXX/drivers/wanpipe` directory run the following command:

```
./Setup install --protocol=AFT_TE1 --silent --builddir=$SANGOMA_DRV_RPM_BUILD_ROOT  
[ optional command argument]
```

The **[optional command argument]** is not mandatory to successfully run the above command. The only supported arguments for the Current Netborder Express version are:

- **--with-linux:** This argument can be used to specify the absolute path that points to a configured (.config file must exist) kernel source. If path is not supplied, this argument is defaulted to `/lib/modules/<kernel_version>/build` where `<kernel_version>` is obtained using the `uname -r` command.

Usage Example:

```
./Setup install --protocol=AFT_TE1 -silent -builddir=$SANGOMA_DRV_RPM_BUILD_ROOT -  
with-linux=<absolute_kernel_source_path>
```

4. Once the command execution succeeds, customize the `$SANGOMA_DRV_RPM_BUILD_ROOT` directory.

eg: Copy a custom `wanpipe1.conf` file into `$SANGOMA_DRV_RPM_BUILD_ROOT/etc/wanpipe` directory, so that the RPM installation can have a configured wanpipe card.

5. Run the following command: `../rpm/generate-drivers-rpm-package.sh`.

6. Once the command execution at step 5 succeeds an RPM package named `wanpipe-<kernel_version>-<wanpipe_version>.rpm` is generated under `/usr/src/redhat/RPMS/i386`.

7. To install Netborder Express Hardware drivers on a target system, just copy the *.rpm* package generated at step 6. Then, from the directory where you've copied that package, run the following command: *rpm -ivh --force <rpm name>*, where *<rpm name>* is the name of the package generated at step 6.

Note that the target system **MUST** have the same kernel as the one specified while generating the rpm package (*--with-linux* argument value if any provided or the same kernel version as the system on which the package was built).