NUT Packager and Integrators Guide

Arnaud Quette

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2.8.0.1	2023-03-01	Current release of Network UPS Tools (NUT).	
2.6.0	2011-01-14	First release of AsciiDoc documentation for Network UPS Tools (NUT).	

Contents

1 Introduction **1**

2 Packagers involved **2**

3 Possible use cases **3**

4 Optimized packaging proposal **3**

4.1 Overview of the package tree . . . 3

4.2 Detailed view of the package tree . . . 4

4.2.1 nut . . . 4

4.2.2 libupsclient1 . . . 5

4.2.3 libupsclient1-dev . . . 5

4.2.4 nut-cgi . . . 5

4.2.5 nut-scanner . . . 5

4.2.6 nut-powerman-pdu . . . 5

4.2.7 nut-snmp . . . 6

4.2.8 nut-xml . . . 6

4.2.9 nut-ipmi . . . 6

4.2.10 nut-modbus . . . 6

4.2.11 nut-linux-i2c . . . 6

4.2.12 nut-macosx-ups . . . 6

4.2.13 nut-clients . . . 7

4.2.14 python-pynut . . . 7

4.2.15 python-nut-gui . . . 7

4.2.16 nut-doc . . . 7

Abstract

The aim of this document is to describe the best way to package the Network UPS Tools, and the best practices across the various packaging implementation of NUT.

So as to these can be spread on all supported platforms as a standard, and as a foundation block to build upon.

Warning
this is a Work In Progress document.

1 Introduction

Packaging is a final aim for software. It eases and completes the software integration into an OS, and allows users to have an easy software installation and support out of the box.

Note
making NUT packaging more uniform should help its documentation, support and maintenance across the supported OSes.

```
*sandbox*
This document assumes that you have read the other NUT documents such
as INSTALL.nut, FAQ, config-notes.txt, config-prereqs.txt ...

Facts about NUT packaging
=========================

NUT has so much evolved those two last years (with USB and SNMP
support, the premises of libraries, ...) that the simple
"1, 2 or 3  package(s)" approach is no more suitable.

This fact has reached a high level since NUT 1.4. Actually,
doing this would result in either being forced to install hosts
of unneeded dependencies (net-snmp, gd, ... as seen on SUSE),
to have a partially broken package [1] or not being able to
use all NUT features [2].

Let's now have an overview on how NUT is currently packaged:

        1) Debian:
        http://packages.qa.debian.org/n/nut.html
                nut, nut-dev, nut-usb, nut-snmp, nut-xml, nut-cgi, nut-doc

        2) Mandriva
        http://cvs.mandriva.com/cgi-bin/cvsweb.cgi/SPECS/nut/
                nut-server
                nut
                nut-cgi

        3) SUSE / Novell
                nut

        4) RedHat

        5) PLD
        http://cvs.pld-linux.org/cgi-bin/cvsweb/SPECS/nut.spec

        ... (FreeBSD, Gentoo Linux, IRIX, NetBSD, OpenBSD)

This shows how much the packages name split is now scattered.
The result is:
- that a user of several systems will be lost, and will
waste time
```

```
- there is a big waste of energy
- this makes things hard to create standard configuration
wizards

[1] NUT build on Debian GNU/Linux m68k and Hurd was once
broken due to hiddev dependencies, and usb support still
included in the core package.

[2]
- snmp-ups driver is not available under Mandrake GNU/Linux,
but its man is present. See
http://rpms.mandrakeclub.com/rpms/mandrake/9.1/i586/Mandrake/RPMS/nut-server-1.2.1-4mdk. ↩
    i586.html
- secured ssh network mode not available (due to deps and/or non free)
- some systems don't provide libupsclient lib/header/.pc so as to
client application (such as wmnut) can't be built
- the logger function is not (well) used, same goes for the syslog (triple
redundancy in Mandriva)
- the solution is partial in every system: lost of tests
case / feedback could be shared
...
```

2 Packagers involved

The following packagers are working on this subject:

- Debian (and derivatives): Arnaud Quette <aquette@debian.org>
- SUSE/Novell: Stanislav Brabec <sbrabec@suse.cz>
- Solaris, OpenSolaris, OpenIndiana and related illumos distributions: Jim Klimov <jimklimov@gmail.com>
- MacOS: Charles Lepple

Note
the people below should be contacted to (re)launch discussions!

The following packagers should be interested in working on this subject:

- FreeBSD: Thierry Thomas? <>
- Mandriva: Oden Erikson? <>
- RedHat / Fedora Core: <>
- Gentoo: <>
- NetBSD: <>
- OpenBSD: <>
- PLD: Andrzej Zawadzki <zawadaa@wp.pl>
- E-Smith: Charlie Brady <charlieb-nut-upsdev@e-smith.com>
- Windows: check with WinNUT author?!
- HP-UX: <>
- IBM AIX: <>

3 Possible use cases

- standalone (1 system + 1-n UPS)
- network server (same as standalone, but serving data to network clients)
- network monitoring client
- network supervision client

TO BE COMPLETED...

4 Optimized packaging proposal

Note
The below proposed packages split is subject to discussion.

The aim of this is to:

- rationalize split according to the above use cases,
- share resources (descriptions, i18n, ...)
- find the best compromise between available features and dependencies,
- standardize nut packages name,
- create the foundation for the upcoming and underway improvements,
- improve nut integration,
- ease and improve user experience.

This standard was created by:

- capitalizing on the experience of existing packages,
- using and improving the use of all nut features,
- considering upcoming nut changes and improvements,
- working closely with packagers.

4.1 Overview of the package tree

FIXME: make a dependency graph

- nut
- libupsclient1
- libupsclient1-dev
- nut-cgi
- nut-scanner

- nut-powerman-pdu
- nut-snmp
- nut-xml
- nut-ipmi
- nut-modbus
- nut-linux-i2c (platform-dependent)
- nut-macosx-ups (platform-dependent)
- nut-clients
- python-pynut
- python-nut-gui (or nut-control-center or Ultimate NUT Tool...)
- nut-doc

4.2 Detailed view of the package tree

Note

- The **Desc** field represent the package's description, as exposed by the packaging system. Each package's description is composed of a paragraph common to all NUT packages, and a part that is specific to the package. The common part (further referenced by **COMMON DESC**) is:

```
Network UPS Tools (NUT) is a client/server monitoring system that allows
computers to share uninterruptible power supply (UPS) and power ↩
    distribution
unit (PDU) hardware. Clients access the hardware through the server, and ↩
     are
notified whenever the power status changes.
```

- The **Files** field lists the content of the package.
- The mentioned **Size** is a rough estimation of packaged and installed size. This may varies across the systems and architecture, and is based upon the Debian x86 packages.
- The **Deps** field lists the dependencies of the packages. The exact name may vary across the various systems.
- The **Comment** field is used to place comment for points subject to discussion.

4.2.1 nut

- Desc:
- Files: dummy/serial/USB drivers + upsd + upslog
- Size:
- Deps:

4.2.2 libupsclient1

- Desc:
- Files:
- Size:
- Deps:

4.2.3 libupsclient1-dev

- Desc:
- Files:
- Size:
- Deps:

Note
the "-dev" suffix is to be replaced by "-devel" on RPM based platforms.

4.2.4 nut-cgi

- Desc:
- Files:
- Size:
- Deps:

4.2.5 nut-scanner

- Desc:
- Files:
- Size:
- Deps:

Note
hard third-party dependency on `libltdl`; recommends `libsnmp`, `libneon`, and the `libusb` variant (0.1 or 1.0) it was built against.

4.2.6 nut-powerman-pdu

- Desc:
- Files:
- Size:
- Deps:

4.2.7 nut-snmp

- Desc:
- Files:
- Size:
- Deps:

4.2.8 nut-xml

- Desc:
- Files:
- Size:
- Deps:

4.2.9 nut-ipmi

- Desc:
- Files:
- Size:
- Deps:

4.2.10 nut-modbus

- Desc:
- Files:
- Size:
- Deps:

4.2.11 nut-linux-i2c

- Desc: (platform-dependent)
- Files:
- Size:
- Deps:

4.2.12 nut-macosx-ups

- Desc: (platform-dependent)
- Files:
- Size:
- Deps:

4.2.13 nut-clients

- Desc:
- Files:
- Size:
- Deps:

4.2.14 python-pynut

- Desc:
- Files:
- Size:
- Deps:

4.2.15 python-nut-gui

(or nut-control-center or Ultimate NUT Tool. . .)

- Desc:
- Files:
- Size:
- Deps:

4.2.16 nut-doc

- Desc:
- Files:
- Size:
- Deps:

```
*sandbox*
nut-server
^^^^^^^^^^
                        Desc:
                        Files: dummy/serial/USB drivers + upsd + upslog
                        Size:
                        Deps: nut-client, libusb, libc/ld

                B) nut-snmp

                        Desc:
                        Files: SNMP driver [/ manager ]
                        Deps: nut-server, net-snmp, libc/ld

                C) nut-client

                        Desc: don't force to have the server part/deps if not needed
```

```
                        Files: upsmon, upsc, upscmd, upsrw + driver.list [+nut-dev (lib, .h ↩
                            , .pc, man]
                        Deps: libc/ld

                E) nut-cgi

                        Deps:
                        Files: snmp-ups and powernet + manpages

                F) nut-doc:

                        Deps:
                        Files: dummycons + manpage

                G) nut-dev:

                        Deps:
                        Files: upsmon, upsc, upscmd, upsrw

                H) nut-scanner:

                        Deps: hard dependency on `libltdl`; recommends `libsnmp`, `libneon ↩
                            `, and the `libusb` variant (0.1 or 1.0) it was built against.
                        Files: nut-scanner tool and libnutscan + manpages


Note: "nut" can be a meta package

This kind of tree obviously needs modification on the conf/make
files of NUT to allow build/install in a separate way.


...
TO BE CONTINUED
...

Configuration option
^^^^^^^^^^^^^^^^^^^^

Example:

name= "ups" or "nut"
./configure \
            --prefix=/ \
            --sysconfdir=/etc/$name \
            --mandir=/usr/share/man \
            --libdir=/usr/lib \
            --includedir=/usr/include \
            --datadir=/usr/share/$name \
            --with-statepath=/var/run/nut \
            --with-altpidpath=/var/run/nut \
            --with-drvpath=/lib/nut \
            --with-cgipath=/usr/lib/cgi-bin/$name \
html-path
            --with-pidpath=/var/run/$name \
            --with-user=$name \
            --with-cgi \
            --without-ssl
...

NOTE: For packaging (OS distribution or in-house) it is recommended to
primarily `./configure --with-all` and then excise `--without-something`
```

```
explicitly for items not supported on your platform, so you do not miss
out on new NUT features as they come with new releases. Some may require
that you update your build environment with new third-party dependencies,
so a broken build of a new NUT release would let you know how to act.
```