Network UPS Tools Release Notes

Russell Kroll, Arnaud Quette, Arjen de Korte, Charles Lepple and Jim Klimov

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2.8.1	2023-10-31	Some changes to API, docs and recipes, in particular to simplify local builds and tests (e.g. to help end-users check if current NUT codebase trunk has already fixed an issue they see with a packaged installation). Revived NUT for Windows effort, further improved other OS integrations. NUT became reference for "UPS management protocol", Informational RFC 9271. Documentation files refactored to ease maintenance. More drivers and new driver categories introduced.	JK
2.8.0	2022-04-26	Change of maintainer. Many changes to API, docs (both style and content), and recipes, with a stress on non-regression test-ability, run-time debug-ability, general codebase maintainability, as well as OS integrations (notably nut-driver-enumerator for systemd and SMF service instance maintenance). Added a lot in area of CI support and documented pre-requisite package lists for numerous platforms, and CI agent set-up. Added libusb-1.x support and many new driver categories (and drivers), and daisychain device connection support. Instant commands enhanced with TRACKING to enable protocol-based waiting for completion of a particular INSTCMD or SET operation.	JK
2.7.4	2016-03-09	NUT variables namespace updated, in particular for outlet groups, alarms and thresholds, ATS devices, and battery.charger.status to supersede CHRG and DISCHRG flags published in ups.status readings. NUT network protocol extended with NUMBER type; some API changes.	AQ
2.7.3	2015-04-22	Documentation revised, including some API changes. Added NUT DDL links. NUT variables namespace updated.	AQ
2.7.2	2014-04-17	The nut-website project was offloaded into a separate repository. FreeDesktop HAL support was removed (obsoleted in GNOME consumer). Introduced nutdrv_atcl_usb driver.	AQ

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2.7.1	2013-11-19	NUT source codebase migrated from SVN to Git (and from Debian infrastructure to GitHub source code hosting). jNut binding split into a separate project. Introduced libnutclient (C++ binding), al175, apcupsd-ups and nutdrv_qx drivers, Mozilla NSS support for simpler licensing than OpenSSL, and a newer apcsmart implementation. Documentation support enhanced with a spell checker, contents massively updated to reflect project changes.	CL
2.6.5	2012-08-08	New macosx-ups driver, new implementation of mge-shut driver. NUT variables namespace updated. Docs cleaned up and revised.	AQ
2.6.4	2012-05-31	New NUT network protocol commands (LIST CLIENTS, LIST RANGE and NETVER), and socket protocol commands (ADDRANGE, DELRANGE). NUT variables namespace updated. Introduced nut-recorder tool.	AQ
2.6.3	2012-01-04	No substantial changes to documentation.	AQ
2.6.2	2011-09-15	Introduced nut-scanner tool and nut-ipmipsu driver, systemd support, and a new apcsmart implementation.	AQ
2.6.1	2011-06-01	Introduced default.* and override.* optional settings in ups.conf, an ups.efficiency report, and outlet.0 special handling.	AQ
2.6.0	2011-01-14	First release of AsciiDoc documentation for Network UPS Tools (NUT).	AQ

Contents

1 Introduction **1**

2 NUT Release Notes (and other feature details) **1**

2.1 Release notes for NUT 2.8.1 - what's new since 2.8.0 . . . 1
2.2 Release notes for NUT 2.8.0 - what's new since 2.7.4 . . . 8
2.3 Release notes for NUT 2.7.4 - what's new since 2.7.3 . . . 14
2.4 Release notes for NUT 2.7.3 - what's new since 2.7.2 . . . 16
2.5 Release notes for NUT 2.7.2 - what's new since 2.7.1 . . . 17
2.6 Release notes for NUT 2.7.1 - what's new since 2.6.5 . . . 18
2.7 Release notes for NUT 2.6.5 - what's new since 2.6.4 . . . 19
2.8 Release notes for NUT 2.6.4 - what's new since 2.6.3 . . . 19
2.9 Release notes for NUT 2.6.3 - what's new since 2.6.2 . . . 21
2.10 Release notes for NUT 2.6.2 - what's new since 2.6.1 . . . 22
2.11 Release notes for NUT 2.6.1 - what's new since 2.6.0 . . . 23
2.12 Release notes for NUT 2.6.0 - what's new since 2.4.3 . . . 24
2.13 Release notes for NUT 2.4.3 - what's new since 2.4.2 . . . 25
2.14 Release notes for NUT 2.4.2 - what's new since 2.4.1 . . . 25
2.15 Release notes for NUT 2.4.1 - what's new since 2.4.0 . . . 26
2.16 Release notes for NUT 2.4.0 - what's new since 2.2.2 . . . 26
2.17 Release notes for NUT 2.2.2 - what's new since 2.2.1 . . . 27
2.18 Release notes for NUT 2.2.1 - what's new since 2.2.0 . . . 27
2.19 Release notes for NUT 2.2.0 - what's new since 2.0.5 . . . 28
2.20 Release notes for NUT 2.0.5 - what's new since 2.0.4 . . . 29
2.21 Release notes for NUT 2.0.4 - what's new since 2.0.3 . . . 30
2.22 Release notes for NUT 2.0.3 - what's new since 2.0.2 . . . 30
2.23 Release notes for NUT 2.0.2 - what's new since 2.0.1 . . . 31
2.24 Release notes for NUT 2.0.1 - what's new since 2.0.0 . . . 31
2.25 Release notes for NUT 2.0.0 - what's new since 1.4.x . . . 32
2.26 Release notes for NUT 1.4.0 - what's new since 1.2.x . . . 34
2.27 Release notes for NUT 1.2.2 - what's new since 1.2.1 . . . 36
2.28 Release notes for NUT 1.2.1 - what's new since 1.2.0 . . . 37
2.29 Historic releases . . . 37

3 NUT Upgrading Notes 38

3.1 Changes from 2.8.0 to 2.8.1 . . . 38
3.2 Changes from 2.7.4 to 2.8.0 . . . 40
3.3 Changes from 2.7.3 to 2.7.4 . . . 41
3.4 Changes from 2.7.2 to 2.7.3 . . . 42
3.5 Changes from 2.7.1 to 2.7.2 . . . 42
3.6 Changes from 2.6.5 to 2.7.1 . . . 42
3.7 Changes from 2.6.4 to 2.6.5 . . . 42
3.8 Changes from 2.6.3 to 2.6.4 . . . 42
3.9 Changes from 2.6.2 to 2.6.3 . . . 43
3.10 Changes from 2.6.1 to 2.6.2 . . . 43
3.11 Changes from 2.6.0 to 2.6.1 . . . 43
3.12 Changes from 2.4.3 to 2.6.0 . . . 43
3.13 Changes from 2.4.2 to 2.4.3 . . . 43
3.14 Changes from 2.4.1 to 2.4.2 . . . 43
3.15 Changes from 2.4.0 to 2.4.1 . . . 43
3.16 Changes from 2.2.2 to 2.4.0 . . . 43
3.17 Changes from 2.2.1 to 2.2.2 . . . 44
3.18 Changes from 2.2.0 to 2.2.1 . . . 44
3.19 Changes from 2.0.5 to 2.2.0 . . . 44
3.20 Changes from 2.0.4 to 2.0.5 . . . 44
3.21 Changes from 2.0.3 to 2.0.4 . . . 44
3.22 Changes from 2.0.2 to 2.0.3 . . . 44
3.23 Changes from 2.0.1 to 2.0.2 . . . 45
3.24 Changes from 2.0.0 to 2.0.1 . . . 45
3.25 Changes from 1.4.0 to 2.0.0 . . . 45
3.26 File trimmed here on changes from 1.2.2 to 1.4.0 . . . 45

4 Project history 46

4.1 Prototypes and experiments . . . 46
4.1.1 May 1996: early status hacks . . . 46
4.1.2 January 1997: initial protocol tests . . . 46
4.1.3 September 1997: first client/server code . . . 46
4.2 Smart UPS Tools . . . 47
4.2.1 March 1998: first public release . . . 47
4.2.2 June 1999: Redesigned, rewritten . . . 47
4.3 Network UPS Tools . . . 48
4.3.1 September 1999: new name, new URL . . . 48
4.3.2 June 2001: common driver core . . . 48

4.3.3 May 2002: casting off old drivers, IANA port, towards 1.0 . . . 48
4.4 Leaving 0.x territory . . . 49
4.4.1 August 2002: first stable tree: NUT 1.0.0 . . . 49
4.4.2 November 2002: second stable tree: NUT 1.2.0 . . . 49
4.4.3 April 2003: new naming scheme, better driver glue, and an overhauled protocol . . . 49
4.4.4 July 2003: third stable tree: NUT 1.4.0 . . . 50
4.4.5 July 2003: pushing towards 2.0 . . . 50
4.5 Backwards and Forwards Compatibility (NUT v1.x vs. v2.x) . . . 50
4.6 networkupstools.org . . . 51
4.6.1 November 2003: a new URL . . . 51
4.7 Second major version . . . 51
4.7.1 March 2004: NUT 2.0.0 . . . 51
4.8 The change of leadership . . . 51
4.8.1 February 2005: NUT 2.0.1 . . . 51

1 Introduction

The primary goal of the Network UPS Tools (NUT) project is to provide support for Power Devices, such as Uninterruptible Power Supplies, Power Distribution Units and Solar Controllers.

This document intends to report high-level changes delivered by NUT project releases, as well as practical nuances for packagers and end-users who would be upgrading their NUT deployments.

It DOES NOT intend to detail the change log: it is very large and complicated to render properly, so is better served by another document artifact.

If you wish to discover how everything came together, have a look at the Project History.

2 NUT Release Notes (and other feature details)

If you're upgrading from an earlier version, see the UPGRADING.adoc file.

Please note that web and source document links, product and service names listed in historic entries of past releases may no longer be relevant.

For a complete and more detailed list of changes, please refer to the ChangeLog file (generated for release archives), or to the Git version control history for "live" codebase.

2.1 Release notes for NUT 2.8.1 - what's new since 2.8.0

https://github.com/networkupstools/nut/milestone/8

- "UPS management protocol", Informational RFC 9271 published by IETF at https://www.rfc-editor.org/info/rfc9271 and the IANA port number registry was updated accordingly at https://www.iana.org/assignments/service-names-port-numbers/-service-names-port-numbers.xhtml?search=3493 (even though this RFC is not formally an Internet Standard)
- NUT documentation files were rearranged, renaming some to `*.adoc` pattern to facilitate automatic rendering in GitHub and IDE GUIs, and adding recipe support for GitHub issue/PR links. This `NEWS` file is now proper asciidoc rendered into `release-notes.pdf` (and HTML versions). [issue #1953, PR #2048] Internally, the documents would use a new way to define cross-linking to other pages and their chapters, to facilitate different renderers (including GitHub UI), and file names created for "chunked HTML" documentation format will no longer have the "chapter number, section number" format which is not easy to maintain over time with independent builds of documentation in NUT and the actual and historic snapshots for nut-website for example. Chapter/Section names will be adapted to produce "chunked HTML" file names instead. Documentation links rendered in GitHub UI should point to the HTML pages served by a current iteration of the NUT website. [PR #226, PR #669]
- A new `configure --enable-spellcheck` toggle should add spelling checks to `make check` (by default, if tools are available) to facilitate quicker acceptance of contributions. [#2067]
- Published a new maintainer GPG key to sign tags and release artifacts, and possibly git commits as well, as part of solution for issue #1410. You can pull it from common OpenPGP servers with the following command:

```
:; gpg --recv-key DE0184DA7043DCF7
gpg: key DE0184DA7043DCF7: public key "Jim Klimov (Doing FOSS
          since last millennium) <jimklimov@gmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

  as part of https://github.com/networkupstools/nut/issues/1410 solution.
- Bug fixes for fallout possible due to "fightwarn" effort and other evolution in NUT v2.8.0 release:

  - The `upsdebugx()` and similar methods were converted to macros in #685 to avoid useless data manipulations and requests for logged information, whose results would be ignored instantly because the debug level is too low. As issue #1455 and PR #1495 found, in two cases the called commands did "meaningfully" modify data — so without debug logs the program misbehaved. A known regression for `upscode2` driver; might be or not be a problem with `upsd` server in NUT v2.8.0 release, fixed for NUT v2.8.1.
  - A table in `cyberpower-mib` (for `snmp-ups` driver) sources was arranged in NUT v2.8.0 release in a way that precluded the driver logic from looking at all of its entries. Also a fix for instant command definitions had in fact broken them due to other development. Regressions fixed for NUT v2.8.1 [#1432, #2029]
  - A change for file-change detection in `dummy-ups` driver for NUT v2.8.0 release misfired on some platforms. Regression fixed for NUT v2.8.1 [#1420]
  - Fixed building of NUT man pages when just a few drivers are selected by `configure` script for custom builds [#1467]
  - Now that `upsdrvctl` can pass debugging level through to the launched driver(s), they would by default stay in the foreground. This can complicate (or simplify, when intentional) the management of service instances. Now there are explicit `upsdrvctl` options for choosing this (`-F`/`-B`), although default behavior is retained. Note that explicit foregrounding mode also keeps `upsdrvctl` tool from exiting and would not wait for one driver to complete initialization before starting another in case of mass-management loop to start all drivers (without specifying the single device) [#1759, #1806, #1875]
  - The `apcsmart` and `apcsmart-old` handled invalid data too zealously and aborted instead of skipping over it, like they did before [#2015]
  - A bit maths optimization in `riello_ser` and `riello_usb` misfired [#2137]
  - Something about compile-time macros or other warnings-related refactoring seems to have confused the MGE SHUT (Serial HID UPS Transfer) driver support [#2022]
  - Some warnings were not detected by the tools or build scenarios used earlier, and only got addressed now

- An issue was identified which could cause `libupsclient` parser of device and host names to crash upon bad inputs (e.g. poorly resolved environment variables in scripts). Now it should fail more gracefully [#2052]
- New `configure --enable-inplace-runtime` option should set default values for `--sysconfdir`, `--with-user` and `--with-group` options to match an existing NUT deployment — for users who are trying if a custom build of recent codebase solves their practical issues. For "quick tests", a shortcut operation `./ci_build.sh inplace` was added [#1714]
- State tree structure and methods (including "dstate" wrapper for common driver internals) was enhanced with time-stamping of last modification (setting, changing, deleting the value or some fields in an entry): this allows to detect stale information in a centralized fashion [#2010]
- We lacked log information about changes of chroot jail (uncommon) and of UID/GID (everywhere), which makes troubleshooting harder (e.g. lack of access to config files or USB device nodes). Now we have it [#1694]
- A `NUT_DEBUG_PID` envvar (presence) support was added to add current process ID to tags with debug-level identifiers. This may be useful when many NUT daemons write to the same console or log file. [#2118]
- huawei-ups2000 is now known to support more devices, noted in docs and for auto-detection [#1448, #1684]
- nutdrv_qx updates:
  - a `battery_voltage_reports_one_pack` driver option was added for devices which "natively" report a `battery.voltag` for a single battery pack or cell, not for the whole assembly [#1279]
  - the `voltronic_qs_protocol` should now accept both "V" (as before) and newly "H" dialects, which otherwise seem interchangeable [#1623]
  - the `armac` subdriver was enhanced to support devices with a different response pattern than previously expected per initial contribution. It was tested to work with Vultech V2000 and Armac PF1 series. [#1978]
- nutdrv_qx and blazer updates:
  - extended default ranges for max battery voltage when guessing [#1279]

- sms_ser, a driver for SMS Brazil UPS Protocol 1Phase, was introduced. NOTE: it may later become a subdriver under nutdrv_qx. [#2090]
- usbhid-ups updates:
  - added support for `subdriver` configuration option, to select the USB HID subdriver for the device manually where automatic match does not suffice (e.g. new devices for which no `vendorid`/`productid` pair was built into any driver, or for different-capability devices with same interface chips, notably "phoenixtec/liebert" and "mge") [#1369]
  - cps-hid subdriver now applies same report descriptor fixing logic to devices with ProductID 0x0601 as done earlier for 0x0501, to get the correct output voltage data [#1497]
  - apc-hid subdriver now also supports ProductID 0x0004 [#1429]
  - ever-hid subdriver reported a `powerfactor` without a namespace (bug in 2.8.0 release), fixed to `outlet.powerfactor`
  - the `usbhid-ups` driver should now reconnect if `libusb` returned a memory allocation error [#1422] (seen as "Can't retrieve Report 0a: Resource temporarily unavailable"), which can cause practical problems in the field — the driver otherwise interpreted the situation as `ups.status` being `OL OFF` and cut the power supply.
  - powercom-hid subdriver: fixed `UPS.Battery.ManufacturerDate` to map to `battery.mfr.date` (not `battery.date` which is the maintenance replacement date) [#1644]
  - added `onlinedischarge_calibration` option for UPSes that report `OL+DISCHRG` when they are in calibration mode [#2104]
- powercom driver should now try harder to refresh data from device [#356]
- tripplite_usb driver now supports configuration of `upsid` to match the specific device (not all firmware/hardware models support this) [#2075]
- apcupsd-ups:
  - improvement for `POLL_INTERVAL_MIN` from PR #797 was buggy [#2007]
  - fix to clean obsoleted readings (if any) AFTER getting new info from an `apcupsd` daemon, to avoid the gap when NUT driver knows nothing [#2007]
- apc_modbus driver was introduced, to cover the feature gap between existing NUT drivers for APC hardware and the actual USB-connected devices (or their firmwares) released since roughly 2010, which deprecated standard USB HID support in favor of Modbus-based protocol which is used across the board (also with their network management cards). The new driver can monitor APC UPS devices over TCP and Serial connections, as well as USB with a patched libmodbus (check https://github.com/EchterAgo/libmodbus/commits/rtu_usb for now, PR pending). [#139, #2063]
  - For a decade until this driver got introduced, people were advised to use apcupsd project as the actual program which talks to a device, and NUT apcupsd-ups driver to relay information back and forth. This was a limited solution due to lack of command and variable setting support, as well as relaying of just some readings (just whatever apcupsd exposes, further constrained by what our driver knows to re-translate), with little leverage for NUT to tap into everything the device has to offer. There were also issues on some systems due to packaging (e.g. marking NUT and apcupsd as competing implementations of the same features) which required clumsy workarounds to get both installed and running. Finally, there is a small matter of long-term viability of that approach: last commits to apcupsd sources were in 2017 (with last release 3.14.14 in May 2016): https://sourceforge.net/p/apcupsd/svn/HEAD/tree/
- dummy-ups:
  - Added an `repeater_disable_strict_start` option to disable the driver exiting upon encountering any kind of error at startup (as repeater). This option should allow for collective `upsdrvctl` startup despite individual target UPS to be repeated or `upsd` not having come up yet. [#2132]
  - Revised detection of file path (for "dummy" mode) which misfired under some conditions, and unified several implementations. [#2118]
- NUT for Windows:

  - Ability to build NUT for Windows, last tackled with a branch based on NUT v2.6.5 a decade ago, has been revived with the 2.8.x era codebase [#5]. It is known that at this time some features are not complete, for more details see https://github.com/-orgs/networkupstools/projects/2/views/1
  - Cross-builds of NUT for Windows using Linux and MinGW (and many custom built dependency packages, as documented in the scripts/Windows/README.adoc file) are now regularly tested on NUT CI farm with moderate integration via custom build script `scripts/Windows/build-mingw-nut.sh` [#1489]
  - Semi-native NUT for Windows builds with MSYS2/MinGW x64 environment are now regularly tested on AppVeyor, with the same `ci_build.sh` script and `Makefile` checks as used across the board for local developer builds, Linux/illumos/FreeBSD/OpenBSD/... on dedicated NUT CI farm on Fosshost, and MacOS on CircleCI [#1552]

- snmp-ups updates:

  - Fixed detection for device agents which wrongly return the sysOID value as a string instead of an OID [#1710]
  - Clearer messages about skipping MIBs during driver initialization [#2037]
  - IETF MIB mapping updated for data points where negative readings are invalid [#1558]
  - Added SNMP subdriver "apc-epdu-mib" for APC easy PDU support [#1674]
  - Added SNMP subdriver "eaton-pdu-nlogic-mib" for nLogic (rebranded Eaton) support [#1698]
  - Added SNMP subdriver "hpe-pdu3-cis-mib" for HPE G2 Metered & Switched PDU initial "unitary" support (no daisychain support yet); also note that due to SNMP v1 implementation limitations on this device, you should prefer SNMP v3 to get both read and write rights [#1713]
  - Fixed processing loop for large SNMPv2/SNMPv3 responses where one item in the middle has a type error [#1682]
  - Better manage the slight nuances (especially in `ups.status`) between Eaton UPSs, and rename mibs from `pw` to `eaton_pw_nm2` and from `pxgx_ups` to `eaton_pxg_ups` [#1715]
  - Fixed the long standing "Warning: excessive poll failures" issue, that was tied to non-existent OIDs, not well handled in some parts of the driver [#1716]
  - `baytech-mib.c` subdriver: fixed `baytech_outlet_status_info[]` set of valid outlet status values [#1871]
  - `cyberpower-mib.c` subdriver: support devices which report the shorter Vendor OID as their sysOID, e.g. "CyberPower PowerPanel Personal" [#1997] and support more data points including hardware status alarms [#1982]

- The `bestfortress` driver shutdown handling was fixed to use a non-trivial default timeout [#1820]

- The `optiups` driver only gave accurate voltage information with 120VAC models and assumed a 12V battery when calculating capacity. There is a protocol command that gives a (fixed) voltage which correlates with the voltage selection DIP switches on the back of the UPS, taking into account whether it is a 120 or 240VAC model. Likewise, now the battery capacity fix is applied globally, based on whether or not the battery voltage is greater than 20V. [#2089]

- GPIO drivers [#1855]:

  - Added a new category of drivers, using GPIO interface to locally connected devices (currently limited to 2018+ Linux libgpiod, but its architecture was designed to support more OSes with their equivalents - PRs welcome)
  - `generic_gpio_libgpiod` driver using `libgpiod` backend was added (defaults to be required on Linux, optional on other platforms)

- Added support for `make install` of PyNUT module and NUT-Monitor desktop application [#1462, #1504]

- Regular CI coverage for NUT codebase enhanced with CircleCI running some scenarios on MacOS, might add Windows in the future. Fixed some build issues for MacOS that had crept into NUT v2.8.0 release [#1415, #1421]

- NUT software-only drivers (dummy-ups, clone, clone-outlet) separated from serial drivers in respective Makefile and configure script options [#1446]

- Fixed support for common USB matching options ("vendor", "device", "bus", etc.) for `riello_usb` and `richcomm_usb` [#1763] and updated man pages of all USB drivers using these options to include the same description [#1766]

- Added a "busport" USB matching option (if supported by the hardware, OS and libusb on the particular deployment, it should allow to specify physical port numbers on an USB hub, rather than logical "device" enumeration values, and in turn — this should be less volatile across reboots etc.) [#2043]

- Added an `allow_duplicates` flag for common USB matching options which may help monitor several related no-name devices (although without knowing reliably which one is which... better than nothing) [#1756]
- The `nut-scanner` program should now suggest same configuration fields as those used by common USB matching options in (most of the) drivers, e.g. adding "device" to the generated configuration section [#1790]
- Stuck drivers that do not react to `SIGTERM` quickly are now retried with `SIGKILL` [#1424]
- Each driver should now report its `driver.state` to help readers determine whether it is initializing, reconnecting, or running regular loops [#1767]
- Code which resolves full paths to libraries should now consider the common environment variable `LD_LIBRARY_PATH` as a preferred possible override to built-in paths (note that most operating systems advise against setting this variable unless troubleshooting, although other systems rely on it) [#805]
- Debug information tracing methods like `upsdebugx()` should now be less limited in the sizes of messages that they can print, such as path names that may be quite long. Note that the OS methods manipulating the strings, and receivers such as logging systems, may still impose limits of their own.
- The `nut-scanner` usage and debug printouts now include the loadable library search paths, to help troubleshooting especially in multi-platform builds; pre-filtering of the built-in paths was introduced (to walk only existing and unique directory names) [#317]
- The nut-scanner program was updated to fall back to loading unresolved library filenames, hoping that `lt_dlopen()` implementation on the current platform would find library files better [#805]
- Detection of `libltdl` in `configure` script updated with fallback code to find it on systems that deliver the library to `/usr/local/lib` (e.g. on FreeBSD) [#1577]
- An explicit `configure --with-nut-scanner` toggle was added, specifically so that build environments requesting `--with-all` but lack `libltdl` would abort and require either to install the dependency or explicitly forfeit the tool (some distro packages missed it quietly in the past) [#1560]
- The `nut-scanner` program should now by default warn about serial numbers which do not make much sense (are duplicate, empty, all same character, etc) [#1810]
- Existing openssl-1.1.0 support added for NUT v2.8.0 release was tested to be sufficient without deprecation warnings for builds against openssl-3.0.x (but no real-time testing was done yet) [#1547]
- upslog: Added support for logging multiple devices with one call to the program [#1604]
- Documentation to integrate NUT USB driver startup with `usb_resetter` script has been contributed to `scripts/usb_resette` (the script itself is tracked externally on GitHub), along with a configuration example for Linux+systemd [#1887]
- Some fixes applied to Solaris/illumos packaging and SMF service support [#1554, #1564]
- Some fixes for builds on older OSes with less functional default system shell interpreters - now `autogen.sh` supports a `CONFIG_SHELL` envvar to inject its value into generated `configure` script [#1736]
  - Note that you may have to install additional tools (possibly from third-party FOSS packaging efforts) to prepare and build the NUT codebase, and/or prefer non-default system provided implementations (e.g. to use the XPG4 `grep` with `-E` support on Solaris as detailed in https://github.com/networkupstools/nut/issues/1736 comments)
  - Build environment configuration notes in docs/config-prereqs.txt file refreshed to cover building of current NUT codebase in CentOS 6 [#1804] and Solaris 8 [#1736, #1738]
- `configure` script, reference init-script and packaging templates updated to eradicate `@PIDPATH@/nut` ambiguity in favor of `@ALTPIDPATH@` for the unprivileged processes vs. `@PIDPATH@` for those running as root [#1719]
- `configure` script enhanced: `--with-unmapped-data-points` option allows to build SNMP and USB-HID subdrivers with entries discovered by the scripts which generated them from data walks, but developers did not rename yet to NUT mappings conforming to docs/nut-names.txt standards [#1699]
- PyNUT.py version bumped to 1.5.0 with some improvements:

  - `ListClients()` method fixed (was broken in many ways), and is now CI-tested [#549]
  - `DeviceLogin()` method added (mostly as aid to CI-test `ListClients()` in a practically relevant manner, so far)

- nutclient C++ library:

  - added `listDeviceClients()` and `deviceGetClients(dev)` to `Client` classes, and `Device::getClients()` to match PyNUT capabilities [#549]

- upsclient C library:

  - added support for `NUT_QUIET_INIT_SSL` environment variable to hide the infamous "Init SSL without certificate database" warning [#1662]

- The `upsd.conf` listing of `LISTEN` addresses was previously inverted (the last listed address was applied first), which was counter-intuitive and fixed for this release [#2012]

- The `upsd` configured to listen on IPv6 addresses should handle only IPv6 (and not IPv4-mappings) to avoid surprises and insecurity; it will now warn if a host name resolves to several addresses (and will only listen on the first hit, as before in such cases) [#2012]

- A definitive behavior for `LISTEN *` directives became specified, to try handling both IPv4 and IPv6 "any" address (subject to `upsd` CLI options to only choose one, and to OS abilities). When both address families are enabled, the `upsd` data server will first try to open an IPv6 socket asking for disabled IPv4-mapped IPv6 address support (if the OS honors that), and then an IPv4 socket (which may fail if the IPv6 socket already covers it anyway); in other words, you can end up with one or two separate listening sockets. [#2012]

- sstate (server state, e.g. upsd) should now "PING" drivers also if they last reported themselves as "stale" (and might later crash) so their connections would be terminated if really no longer active [#1626]

- Clarified documentation in codebase according to end-user feedback [#1721, #1750 and others over time]

- upsmon client changes include:

  - Several fixes for `upsmon` behavior [#1761, #1680...], including new ability to configure default POWERDOWNFLAG location — packagers are encouraged to pick optimal location for their distributions (which remains mounted at least read-only late in shutdown) and a new optional POLLFAIL_LOG_THROTTLE_MAX setting [#529, #506]
  - Also `upsmon` should now recognize `OFF` and `BYPASS` flags in `ups.status` and report that these states begin or end. The `OFF` state usually means than an administrative action happened to power off the load, but the UPS device is still alive and communicating (USB, SNMP, etc.); corresponding `MONITOR`'ed amount of power sources are considered not being "fed" for the power value calculation purposes. The `BYPASS` state is now treated similarly to `ONBATT`: currently this UPS "feeds" its load, but if later communications fail, it is considered dead. This may have unintended consequences for devices (or NUT drivers) that do not report these modes correctly (e.g. an APC calibration routine seems to start with a few seconds of "OFF" state), so the reported status is only considered as a loss of feed if it persists for more than `OFFDURATION` seconds. [#2044, #2104]
  - Introduced `SHUTDOWNEXIT no` configuration toggle for systems which require a long time to stop their workload such as virtual machines. Since the disconnection of a "secondary" client is treated by the "primary" system as permission to proceed with its own shutdown and power-off for the UPS, the original (now merely default) behavior to call `SHUTDOWNCMD` and immediately exit could be counter-productive. An optional delay can also be introduced. [#2133]
  - Note there were other changes detailed below which impacted several NUT programs, including `upsmon`.

- Extended Linux systemd support with optional notifications about daemon state (READY, RELOADING, STOPPING) and watchdog keep-alive messages [#1590]

  - Normally **inability** to send such notifications (e.g. lack of systemd or similar framework on the particular platform) would be reported once per daemon uptime on its console log, to help troubleshooting situations where such lack of notifications can cause automated service restarts. These messages can be hidden by setting `NUT_QUIET_INIT_UPSNOTIFY=true` environment variable in init-scripts on platforms where such frameworks are not expected. [#2136]

- Extended Linux systemd units with aliases named after the daemons: `nut-server.service` as `upsd.service`, and `nut-monitor.service` as `upsmon.service` (so simple `systemctl reload upsd` can work) [#1777]

- Extended driver-server socket protocol with `BROADCAST (num)` keyword, and a `NOBROADCAST` as a shortcut for `BROADCAST 0`. This allows clients to toggle whether they want to receive `send_to_all()` updates from a driver, or only answers to requests they send [#1914]

- Added support for `make sockdebug` for easier developer access to the tool; also if `configure --with-dev` is in effect, it would now be installed to the configured `libexec` location. A man page was also added. [#1936]

- Numerous daemons (`upsd`, `upsmon`, drivers, `upsdrvctl`, `upssched`) which accepted `-D` option for debug setting previously, now can also honour a `NUT_DEBUG_LEVEL=NUM` environment variable if no `-D` arguments were provided. Unlike those arguments, the environment variable does not enforce that daemons run in foreground mode by default [#1915]

  - Note that unlike some other NUT daemons, `upssched` with enabled debug does not stop reporting on `stderr`! [#1965]

- A bug in `upssched` was discovered and fixed, where it ran a tight loop stressing the CPU; it was presumably introduced between NUT v2.7.4 and v2.8.0 releases [#1964, #1965]

- Implemented generic support for INSTCMD and SETVAR use-cases shared by all drivers, and in particular to see and change active debug verbosity using the driver-server and server-client protocol (at higher priority than CLI or config file choices) per [#1285], e.g.

```
# Set verbosity level 6:
:; upsrw -s driver.debug=6 UPS

# Set verbosity level 0 to disable the noise (even if debug_min is set):
:; upsrw -s driver.debug=0 UPS1@localhost

# Un-set the protocol override, honour CLI or config-file settings again:
:; upsrw -s driver.debug=-1 remoteUPS@1.2.3.4
```

+ and a `driver.killpower` instant command (for safety, must be unlocked by `driver.flag.allow_killpower` protocol setting or `allow_killpower` configuration flag), which is now the first choice for `driver -k` operations [#1917, #1923]

- Implemented basic support for `ups.conf` reloading in NUT drivers, with a `driver.reload-with-error` instant command (more commands and signal handling may be available depending on platform), with a goal of changing inconsequential settings like `debug_min` for a running driver. This can also benefit the drivers on systems managed by real-time `nut-driver-enumerator` and for simpler changes the drivers get only reloaded and not redefined and restarted. Reload signals should also be reasonably supported with `upsdrvctl` tool. Relevant CLI options for `-c CMD` handing were added to drivers and `upsdrvctl`, although their availability may vary between operating systems [#1903, #1914, #1924]

- Drivers should now accept `SIGURG` (or `SIGWINCH` on systems that lack the former) on POSIX platforms to dump their current state information and move on. Such report goes to `stdout` of the driver process (may be disconnected when background mode is used) — this can help with troubleshooting [#1907]

- Recipes and `main.c` code were enhanced to produce a `libdummy_mockdrv.la` helper library during build (not intended to be installed nor distributed), in order to facilitate creation of test programs which behave like a driver [#1855]

- Further revision of public headers delivered by NUT was done, particularly to address lack of common data types (`size_t`, `ssize_t`, `uint16_t`, `time_t` etc.) in third-party client code that earlier sufficed to only include NUT headers. Sort of regression by NUT 2.8.0 (note those consumers still have to re-declare some numeric variable types used) [#1638, #1615]

- The `COPYING` file was updated with licenses and attribution for certain source code files and blocks coming from the Internet originally [#1758]

- The `tools/gitlog2changelog.py.in` script was revised, in particular to generate the `ChangeLog` file more consistently with different versions of Python interpreter, and without breaking the long file paths in the resulting mark-up text [#1945, #1955]

- The "NUT client for VMware ESXi" project (by René Garcia) got its build recipes published on GitHub at https://github.com/-rgc2000/NutClient-ESXi [#1961]

2.2 Release notes for NUT 2.8.0 - what's new since 2.7.4

Note
Earlier discussions (mailing list threads, GitHub issues, etc.) could refer to this change set (too long in the making) as NUT 2.7.5.

- New (optional) keywords for configuration files were added, so existing NUT 2.7.x builds would not accept them if some deployments switch versions back and forth — due to this, semantically the version was bumped to NUT 2.8.x.
- Add support for openssl-1.1.0 (Arjen de Korte)
- libusb-1.0 API support in addition to libusb-0.1 API [#300]
- Add support for `DISABLE_WEAK_SSL=true` in upsd.conf to disable older/weaker SSL/TLS protocols and ciphers: when NUT is built against relatively recent versions of OpenSSL or NSS it will be restricted to TLSv1.2 or better. For least-surprise, currently defaults to `false` and complains in log [PR #1043]
- Add support for `ALLOW_NO_DEVICE=true` (as an upsd.conf flag or environment variable passed from caller of the program), to allow starting the data server initially without any device configurations and reloading it later to apply config changes on the fly [PR #766]
- Add support for `debug_min=NUM` setting (ups.conf, upsd.conf, upsmon.conf) to specify the minimum debug verbosity for daemons. This allows "in-vivo" troubleshooting of service daemons without editing init scripts or service unit definitions.
- Improve support for upsdrvctl for managing of numerous device configs, including default "maxretry=3" and a "nowait" option to complete the "start of everything" mode after triggering the drivers and not waiting for them to complete initializing. This matters on systems that monitor from dozens to hundreds of devices.
- Drivers support a new value for `synchronous` setting, which is the new default now: `auto`. Initially after driver start-up this mode acts as the older default `off`, but would fall back to `on` in case the driver fails to send reports to `upsd` by overflowing the socket buffer in async mode — so the next connections of this driver uptime would be synchronized (potentially slower, but safer — blocking on writes to the data server). This adaptation would primarily impact and benefit devices with many (hundreds of) data points, such as ePDUs and daisy chains. [issue #1309, PR #1315]
- Daemons such as upsd, upsmon, upslog, and device drivers previously implied that enabled debugging (or upslog to stdout) means foreground running, otherwise the daemon was always sent to the background. Now there are explicit options for this (`-F`/`-B`), although default behavior is retained. This change is used for simplified service unit definitions.
- Improvements for device discovery or driver "lock-picking", including general support for:
  - "Standalone" mode (`-s` option), to monitor a device which is not detailed or mentioned in ups.conf
  - `NUT_ALTPIDPATH` and `NUT_STATEPATH` environment variables to override the paths built into the driver binary [PR #473 and #507]
  - "Driver data dump" mode (`-d` option), to poll a device for one or few (*update_count*) loops, report discovered values (dump the data tree in upsc-like format), and exit. This complements the `nut-scanner` for finding and identifying devices.
- support for new devices:
  - IBM 6000 VA LCD 4U Rack UPS; 5396-1Kx (USB)
  - Phoenix Contact QUINT-UPS model 2320461 (Modbus)
  - Tripp-Lite SU3000LCD2UHV (USB; protocol 1330)
  - Emerson Avocent PM3000 PDU (SNMP)
  - HPE ePDU (SNMP)
- nutdrv_qx: enhanced estimation of remaining battery runtime based on speed of voltage drop, which varies as they age [PR #1027]
- nutdrv_qx: several subdrivers added or improved, including:

  - "snr" subdriver with USB connection, for SNR-UPS-LID-XXXX [PR #1008]. Note that end-users should reference explicitly the `snr` subdriver in their `ups.conf` settings because of USB chip using the same values of VendorID/ProductID as fabula_subdriver, fuji_subdriver, and krauler_subdriver.
  - "hunnox" subdriver, as a dialect of earlier "fabula" [PR #638] adds support for Hunnox HNX-850 with USB connection and reported to work for Powercool, Iron Guardian, ARES devices and possibly many others from discussions linking to the pull request which introduced the driver.
  - "phoenixtec" subdriver for Masterguard A and E series, device series A700/1000/2000/3000(-19) and E40/60/100(-19). [PR #975]
  - "ablerex" subdriver provided by the OEM vendor, note that it replaces "krauler_subdriver" as default handler for VID:PID 0xffff:0x0000 [PR #1135]
  - Legrand HID defined and handled by "krauler_subdriver" by default [PR #1075, issue #616]
  - add new "armac" subdriver, tested with Armac R/2000I/PSW, but should support other UPSes that work with "PowerManagerII" software from Richcomm Technologies from around 2004-2005 [PR #1239, issue #1238]

- microsol-apc (starting at version 0.68 as derived from solis 0.67): adding support for newer APC Back-UPS BR hardware, such as APC Back-UPS BZ1500, BZ2200BI and BZ2200I [PR #994]
- pijuice: added new i2c bus driver for PiJuice HAT, a battery UPS module for the Raspberry Pi systems [PR #730]
- huawei-ups2000: added new driver for USB (Linux 5.12+ so far) and Serial RS-232 Modbus device support of Huawei UPS2000/2000A (1kVA-3kVA) series, and possibly some related FSP UPS models. [PR #954]
- socomec_jbus: added new driver for modbus-based JBUS protocol over serial RS-232 for Socomec UPS (tested with a DIGYS 3/3 15kVA model, working on Linux x86-64 and Raspberry Pi 3 ARM). [PR #1313]
- adelsystem_cbi: added new driver for ADELSYSTEM CBI2801224A, an all-in-one 12/24Vdc DC-UPS, which supports the modbus RTU communication protocol [PR #1282]
- generic_modbus: added new driver for TCP and Serial Modbus device support. The driver has been tested against PULS UPS (model UB40.241) via MOXA ioLogikR1212 (RS485) and ioLogikE1212 (TCP/IP), and configuration allows to map custom registers and addresses to NUT events [PR #1052]
- genericups: added support for FTTx battery backup devices, and new signal type mappings for the contact closure pins interpretation (RB for replace battery, BYPASS for disconnected battery, and "none" or NULL for signals to ignore) [PR #1061]
- add devices to HCL/DDL:
  - APC Back-UPS CS (USB)
  - CPS CP1500EPFCLCD (USB)
  - CPS EC350G, EC750G (USB)
  - CPS PR2200LCDRT2U (SNMP)
  - Eaton ATS 16 and 30 (SNMP)
  - Eaton 5E2200VA (USB)
  - Eaton 9PX Split Phase 6/8/10 kVA (XML/USB/SHUT)
  - Eaton 9PX (XML/USB/SHUT)
  - Eaton Ellipse PRO 650 VA (USB)
  - Ippon Back Comfo Pro II 650/850/1050 (USB)
  - Numeric Digital 800 (USB)
  - Opti-UPS PS1500E (USB)
  - Powercool 350VA to 1600VA (USB)
- C++11 support in nutclient library and cppunit tests
- Added C++ testing mock for TcpClient class (nutclientmem/MemClientStub: data stored in local memory) [PR #1034]

- Dual Python 2 and 3 compatibility in development scripts; ability to run build activities and resulting built NUT programs on systems that do not have a binary named "python" [PR #1115 and some before it]
- Added Russian translation for NUT-Monitor GUI client [PR #806]
- Separated NUT-Monitor UI into two applications, NUT-Monitor-py2gtk2 and NUT-Monitor-py3qt5, suitable for two generations of Python ecosystem with their great differences; `NUT-Monitor` name is retained for wrapper script which calls one of these, such that the current system can execute [PRs #1310, #1354]
- Various USB driver families: expanded device-matching with "device" in addition to "bus" and generic USB fields. This is needed to support multiple attached devices that seem identical by other fields (e.g. same vendor, same model, same USB bus, and no serial number) [PR #974]
- Various USB driver families: Improved HID parsing for byte-stream to number conversions on different CPU architectures [PR #1024]
- Various USB HID driver families: added support for composite devices utilizing interface greater than 0 for the UPS interface [PR #1044]
- usbhid-ups:
  - added generic framework for fixing Report Descriptors which can be used for different manufacturers by adding code to the appropriate subdriver rather than polluting the main code with UPS specific exceptions, and applied fixes for known mistakes in (some releases of firmware for) CyberPower CPS*EPFCLCD [issue #439, PR #1245]
  - added `onlinedischarge` option for UPSes that report `OL+DISCHRG` when wall power is lost [PR #811]
  - changed detection of VendorID 0x06da handling of which is claimed by Liebert/Phoenixtec HID historically, and MGE HID (for AEG PROTECT NAS UPSes) since NUT 2.7.4, so that the higher-priority MGE subdriver would not grab each and all of the devices exposing that ID [PR #1357]
  - CPS HID: add input.frequency and output.frequency
  - OpenUPS2: only check OEM Information string once (fewer log messages)
  - Liebert GXT4 USB VID:PID [10AF:0000]
  - add battery voltage and input/output transfer voltage and frequency in Liebert/Phoenixtec HID mapping, to support PowerWalker VFI 2000 TGS better [PR #564, issue #560]
  - add a little delay between multicommands [PR #1228]
  - fix Eaton/MGE mapping for beeper handling
  - add IBM USB VID
  - add deep battery test for CyberPower OL3000RMXL2U
  - report the libusb version used
  - fixed CPU architecture dependent bitmask math issues, causing wrong numbers interpreted from wire protocol data in Big-Endian LP64 builds (SPARC64, s390x, etc.) [issue #1023, PRs #1024, #1040, #1055, #1226]
  - add Delta UPS Amplon R Series, tested on R1K and R3K model [PR #987]
  - add Delta Minuteman UPS VID/PID [PR #1230, issues #555 and #1227]
  - add AMETEK Powervar UPM [PR #733]
  - add Tripplite AVR750U (ProductID 0x3024) [PR #963]
  - add Arduino HID device support with new arduino-hid subdriver [PR #1044]
  - add new salicru-hid subdriver, tested with Salicru SPS Home 850 VA [PR #1199, issue #732]
  - add new ever-hid subdriver to support EVER UPS devices (Sinline RT Series, Sinline RT XL Series, ECO PRO AVR CDS Series) [PR #431]
  - add ability to set `battery.mfr.date` for APC HID UPS [PR #1318]
- usbhid-ups / mge-shut: compute a realpower output load approximation for Eaton UPS when the needed data is not present
- snmp-ups:

  - APC ePDU MIB support
  - add `input.phase.shift` variable
  - add configurable write-able `ondelay` (`ups.delay.start`) and `offdelay` (`ups.delay.shutdown`) as timeticks support [PR #276]
  - outlet groups
  - fix the rounding / truncation of some values
  - add outlet.N.name for Eaton ePDU
  - add input.bypass.frequency for Eaton 3ph
  - fix support for Eaton 2-phase ("split phase") UPS
  - add flag to list currently loaded MIB-to-NUT mappings
  - fix input.L2.voltage on Eaton G2/G3 PDU
  - update Eaton Aphel Revelation MIB
  - support Raritan Dominion PX2 PDU
  - support Emerson Avocent PM3000 PDU
  - improve ALARM flag handling
  - add firmware version for new HPE Network card
  - add ups.load, battery.charge, input.{voltage,frequency} and output.voltage for CyberPower, as well as shutdown and other instant commands
  - several rounds of updates for Eaton devices, including new ATS and ePDU hardware families
  - fixed bit mask values for flags to surely use different numbers behind logical items (inevitably changing some of those macro symbols) [PR #1180]
- snmp-ups and nut-scanner should now support more SNMPv3 Auth and Priv protocols, as available at NUT build time [PRs #1165, #1172]
- nut-scanner: various improvements, including:
  - detection of libraries at runtime
  - tracing information
  - limiting parallelism (thread count) [PRs #1158, #1164]
- nut-ipmipsu: improve FreeIPMI support to build cleanly against older and newer FreeIPMI versions [PR #1179]
- the powerpanel driver now also supports CyberPower OR1500LCDRTXL2U with serial cable [PR #538]
- powercom driver: implement `nobt` config parameter to skip battery check on initialization/startup [PR #1256]
- netxml-ups:
  - Report calibration status
  - Fix for erroneous battery info (MGEXML/0.30) [PR #1069]
- solis: various improvements and fixes
- liebert-esp2: Correct battery V scaling, update docs, implement split-phase unit support [PR #412]
- tripplite: the "Tripp-Lite SmartUPS driver" as tested with SMART2200NET learned to discover the firmware generation and some device features, and in particular to manage power separately on one or two outlet groups [PR #1048]
- tripplite_usb: updated to recognize the "3005" protocol [PR #584]
- libnutclient: introduce getDevicesVariableValues() to improve performances when querying many devices (up to 15 times faster)

- nut-driver-enumerator: introduced a script for Linux systemd and Solaris/illumos SMF to inspect current NUT configuration in ups.conf file and generate service management instances for each currently tracked power device. Also introduced services to monitor the NUT configuration and react to editions of this file, mostly intended for deployments that do massive monitoring of dynamically changing farms of power devices.
- Fix File descriptors leaks by upsmon and upssched (SELinux errors)
- systemd support improvements:
  - POWEROFF_WAIT
  - reload support for upsd
  - Deliver systemd-tmpfiles config to pre-create runtime locations [PR #1037 for Issue #1030]
  - Update units with SyslogIdentifier=%N for better logging [PR #1054]
- upsrw: display the variable type beside ENUM / RANGE
- Added `PROTVER` as alias to `NETVER` to report the protocol version in use. Note that NUT codebase itself does not use this value and handles commands and reported errors individually [issue #1347]
- Implement status tracking for instant commands (instcmd) and variables settings (setvar): this allows to get the actual execution status from the driver, and is available in libraries and upscmd / upsrw [PR #659]
- Add support for extra parameter for instant commands, both in library and in upscmd
- dummy-ups can now specify `mode` as a driver argument, and separates the notion of `dummy-once` (new default for `*.dev` files that do not change) vs. `dummy-loop` (legacy default for `*.seq` and others) [issue #1385]
- new protocol variables:
  - `input.phase.shift`
  - `outlet.N.name`
  - `outlet.N.type`
  - `battery.voltage.cell.max`, `battery.voltage.cell.min`
  - `battery.temperature.cell.max`, `battery.temperature.cell.min`
  - `battery.status`
  - `battery.capacity.nominal`
  - `battery.date.maintenance` (and clarified purpose of `battery.date`)
  - `battery.packs.external` (and clarified purpose of `battery.packs`)
  - `experimental.*` namespace introduced [PR #1046] to facilitate introduction of NUT drivers and their data points for which we do not yet have concepts, or which the original driver contributors did not map well per suitable NUT standards: this allows to balance having those drivers available in the project vs. least surprise for when the explicitly experimental names are changed to something stable and standardized.
  - Proposed to track Date and Time values (still as "opaque strings") preferably in representations compatible to ISO-8601/RFC-3339 [PR #1076] (standards update; changes to actual codebase to be applied in the future)
    - New routine to convert a US formatted date string "MM/DD/YYYY" to an ISO 8601 Calendar date "YYYY-MM-DD" was added to snmp-ups.c [PR #1078]
- Master/Slave terminology was deprecated in favor of Primary/Secondary modes of `upsmon` client:
  - Respective keywords in the configuration files (`upsd.users` and `upsmon.conf`) are supported as backwards-compatible settings, but the obsoleted values are no longer documented.
  - Protocol keyword support was similarly updated, with `upsmon` now first trying to elevate privileges with `PRIMARY <ups>` request, and falling back to `MASTER <ups>` just in case it talks to an older build of an `upsd` server.
  - For the principle of least surprise, NUT codebase still exposes the `net_master()` (as handler for `MASTER` net command) in header and C code for the sake of existing linked binaries, and returns the `OK MASTER-GRANTED` line to the older client that invoked it.

  - Newly introduced `net_primary()` (as handler for `PRIMARY` net command) calls the exact same application logic, but returns `OK PRIMARY-GRANTED` line to the client.
  - Python binding updated to handle both cases, as the only found in-tree protocol consumer of the full-line text.
  - For more details see issue #840 and several pull requests referenced from it, and discussions on NUT mailing lists.

- Build fixes:

  - In general, numerous fixes were applied to ensure portability and avoid warnings (fixing a number of real bugs that caused them); CI was extended to keep the codebase free of those types of warnings which we have got rid of, requiring builds to succeed cleanly in several dozen combinations of compiler versions, C standard revisions (C99 upwards, though on many OSes with GNU99+ extensions), operating systems and CPU architectures.
  - Public CI introduced to automatically test every contribution (PR) and resulting increment of main NUT codebase, including Travis CI and LGTM.com services, and a Jenkins farm on virtual hardware donated by Fosshost.org; this augments testing earlier provided for some branches by Buildbot.
  - Added cppunit testing with valgrind for the C++ client library
  - Make targets added for shell script syntax checks for helper and service scripts
  - Make targets added for spellcheck and for maintenance of the dictionary, including incremental spellcheck to only parse recently edited text files
  - The AsciiDoc detection has been reworked to allow NUT to be built from source without requiring asciidoc/a2x (using pre-built man pages from the distribution tarball, for instance)
  - Makefile contents rearranged for more resilient out-of-tree and in-tree builds beside those made from the root workspace directory
  - Makefiles are tested with GNU Make and BSD Make to ensure portable recipes
  - More use of `pkg-config` to detect dependencies at configure time, as well as fail-safe detection of presence of pkg-config (and its macros) to survive and build without it too
  - "slibtool" pedantic nuances now supported, allowing an alternative to GNU libtool
  - Build scripts updated to remove obsoleted calls to cleanly work with autoconf-2.70 releases in 2020 (also works with 2.69 which was the earlier release since 2012)
  - Dynamic library loading used in certain programs and use-cases improved, especially for 64-bit vs 32-bit builds on multiple-bitness OSes
  - Logging routines like `upsdebugx()` were refactored as macros so there is slightly less overhead when logging is disabled [PRs #685 and #1100]
  - Numerous classes of compilation warnings eradicated, many of those being potential issues with implicit data type conversions and varied numeric type width, signedness, string buffer size, uninitialized variables or structure fields; some more in progress
  - Several logical errors found and fixed during this walk over codebase.
  - Cases where compilers were overly zealous and particular code was written the way wit was intentionally, including some comparisons that help with different-bitness builds but indeed seem superfluous in a certain single bitness, were commented and encased in pragmas to disable the warnings
  - Basic coding style (indentations, lack of trailing white space) applied per developer guide, but not automatically enforced/checked yet.

- Due to changes needed to resolve build warnings, mostly about mismatching data types for some variables, some structure definitions and API signatures of several routines had to be changed for argument types, return types, or both. Primarily this change concerns internal implementation details (may impact update of NUT forks with custom drivers using those), but a few changes also happened in header files installed for builds configured `--with-dev` and so may impact `upsclient` and `nutclient` (C++) consumers. At the very least, binaries for those consumers should be rebuilt to remain stable with NUT 2.8.0 and not mismatch int-type sizes and other arguments.

- As usual, more bugfixes, cleanup and improvements, on both source code and documentation.

2.3 Release notes for NUT 2.7.4 - what's new since 2.7.3

- New class of device supported: ATS - Automatic Transfer Switch are now supported in NUT. Eaton ATS are supported, and APC ones should be too. Users are welcomed to test and provide feedback
- NUT command and variable naming scheme:
  - Document battery.charger.status, which will in time replace the historic CHRG and DISCHRG flags published in ups.status
  - Many extensions to support outlets groups, thresholds / alarms (ambient, input, output, outlet and outlet.group)
- support for new devices:
  - AEG PROTECT B / NAS
  - APC ATS AP7724 (should be supported)
  - Asium P700
  - Eaton ATS
  - Eaton 5E 1100iUSB
  - Eaton E Series DX UPS 1-20 kVA
  - Eaton Powerware 9125-5000g
  - Electrys UPS 2500
  - Fideltronic INIGO Viper 1200
  - Legrand Keor Multiplug
  - LYONN CTB-800V
  - Micropower LCD 1000
  - NHS Laser Senoidal 5000VA
  - Sweex model P220
  - TS Shara
  - Various APCUPSD-controlled APC devices
- snmp-ups:
  - Improve automatic detection algorithm
  - Provide access to Net-SNMP timeout and retries
  - Proper handling of integer RW variables
  - Implement support for alarms, through ups.alarm and outlet.n.alarm
  - Improve log/debug output trace
  - Fix loss of precision when setting values, using upsrw
  - Support for outlets group management
  - Many improvements and simplification
  - Add support for Tripplite units using IETF mib
  - Improve communication staleness detection and recovery
  - Add devices MAC address publication
  - Register values enumerations, when available
  - Many improvements and fixes to the SNMP subdriver creation script
- Eaton:
  - 3ph SNMP: Many improvements to Powerware / XUPS MIB, for data and commands Add support for Eaton Power Xpert Gateway UPS Card Improve support for temperature and humidity, including low / high values Alarms handling

  - ePDU (G2 and G3): Improve support for ambient sensor, including thresholds and dry contacts Outlet groups handling, including data, thresholds, settings and commands Alarms handling
  - XML/PDC (netxml-ups): Fix Eaton XML published data Add some settings (R/W flags) on ambient thresholds
- bcmxcp_usb: improvements for device claiming and multi-packets responses
- dummy-ups: allow any variable to be modified
- libnutclient: Fix for reads when the socket was closed by NUT server
- macosx-ups:
  - fix for 10.10 (Yosemite), v1.1
  - gracefully handle disconnection of UPS (return "data stale")
- nutdrv_atcl_usb: point to nutdrv_qx (fuji) for 0001:0000
- nutdrv_qx:
  - Add new *sgs* USB subdriver to support TS Shara units
  - various improvements and simplification, to the code and documentation
- nut-ipmipsu: improve FreeIPMI support
- nut-scanner:
  - Don't depend on development libraries, by looking at some known paths, including the one provided through --libdir, to find the correct libraries
  - Fix a crash on a 2nd call to libnutscan with SNMP method
- powercom: fix the processing of input and output voltage for KIN units
- solis:
  - many improvements and cleanup
  - resync with end-of-packet character
  - fixes for Microsol Back-Ups BZ1200-BR
- tripplitesu: Fix initialization when tripplite firmware is buggy (for Tripplite SU1000RT2U and possibly more)
- usbhid-ups:
  - various minor improvements
  - support for Eaton UPS with dual HID report descriptor in HID Parser
  - handle missing USB strings in APC code
- SSL support through Mozilla NSS: Rework the NSS tests to ensure that NSS is actually installed and usable for enabling SSL support in NUT
- Augeas support: Augeas lens for ups.conf was updated to add various missing global directives and ups fields
- scripts/systemd/nut-server.service.in: Restore systemd relationship since it was preventing upsd from starting whenever one or more drivers, among several, was failing to start
- Fix UPower device matching for recent kernels, since hiddev* devices now have class "usbmisc", rather than "usb"
- Network protocol information: default to type NUMBER for variables that are not flagged as STRING . This point is subject to improvements or change in the next release 2.7.5. Refer to docs/net-protocol.txt for more information
- As usual, more bugfixes, cleanup and improvements, on both source code and documentation.

2.4 Release notes for NUT 2.7.3 - what's new since 2.7.2

- reverted POWERDOWNFLAG to /etc/killpower as in 2.6.5 (packagers may want to put this in another filesystem, though)
- configure/make fixes for `systemdsystemunitdir`
- apcsmart: fix command set parsing for protocol version 4 (e.g. Smart-UPS RT 10000 XL)
- upslog: SIGUSR1 forces an immediate log entry
- riello_usb/_ser: USB interface claim fix; improved error handling
- usbhid-ups: add support for OpenUPS2 (PID: D005), Liebert GXT3 (PID: 0008) APC AP9584 Serial→USB kit (PID: 0000), and some Powercom models (PID: 0001). Fixed scaling for Cyberpower 0764:0501.
- USB core: do not call usb_set_altinterface(0) by default
- nutdrv_qx:
  - added fabula, fuji USB and Voltronic-QS-HEX subdrivers
  - add bestups subdriver to supersede the old standalone bestups driver
- NUT Monitor: added FreeDesktop AppData file (including screenshots)
- renamed udev rules file to 62-nut-usbups.rules (permissions fix)
- added AIX packaging
- asem: added a driver for the UPS in ASEM PB1300 embedded PCs
- solis: updated to support APC Microsol units sold in Brazil
- tripplite_usb:
  - updated to use dv/dq charge calculation for all models (also exposes battery_min and battery max as configuration variables)
  - added binary 3005 protocol support (such as for SMART500RT1U)
- genericups: better debugging while parsing the cable description flags
- all drivers: a new *synchronous* driver flag is available for very verbose units, such as some ePDUs
- Eaton:
  - Add support for EnergySaving features for Eaton UPSs (HID USB/SHUT and XCP USB/serial)
  - Fix and complete Eaton ePDUs G2/G3 support
  - ABM (Advanced Battery Monitoring) support through battery.charger.status in HID (USB and SHUT), XCP (USB and serial) and SNMP (Powerware XUPS MIB)
- support for new devices:
  - APC Back-UPS 1200BR and Back-UPS BZ2200BI-BR (Microsol)
  - ASEM SPA PB1300 UPS
  - Belkin Regulator PRO-USB
  - Cyber Power Systems Value 1500ELCD-RU
  - EUROCASE EA200N 2000VA
  - Fideltronik LUPUS 500
  - Flight Technic & International (FTUPS) FT-1000BS and FT-1000BS(T)
  - Grafenthal PR-3000-HS
  - JAWAN JW-UPSLC02

  - Lacerda New Orion 800VA
  - Mecer ME-1000-WTU
  - NHS Sistemas de Energia Expert C Online 6000/8000/10000
  - NHS Sistemas de Energia Expert S Online 6000/8000/10000
  - Powercom BNT-xxxAP (USB product id: 0001)
  - Rucelf UPOII-3000-96-EL
  - Tripp Lite OMNIVSINT800
  - Voltronic Power Apex 1KVA and Imperial 1KVA

2.5 Release notes for NUT 2.7.2 - what's new since 2.7.1

- This release is the second interim release of the 2.7 testing series.
- libupsclient had undefined references related to functions of libcommon. This issue was reported on Debian (bug #731156) and is now fixed
- support for new devices:
  - CABAC UPS-1700DV2
  - Eaton Powerware 3105
  - Emerson Network Power Liebert PSI 1440
  - MicroDowell B.Box LP 500
  - Numeric Digital 800 plus
  - OptiUPS VS 575C
  - Tripp Lite SU10KRT3/1X
- FreeDesktop Hardware Abstraction Layer (HAL) support was removed.
- nutdrv_atcl_usb: new driver for *ATCL FOR UPS*
- al175: re-introduced this driver (actually, it was in 2.7.1)
- upsdrvctl now provides retry options for upsdrvctl and driver(s)
- snmp-ups: add support for XPPC-MIB and Tripp Lite SU10KRT3/1X. Also fix erroneous status in HP/Compaq SNMP MIB (with the most recent HP firmware (1.76); improved various MIBs (APC, HP/Compaq, ...)
- nutdrv_qx: add new *fallback* Q1 subdriver, with minimal *Q1* support. General improvements on all subdrivers.
- mge-shut: partially revert PnP/RTS change, for initializing the communication with the UPS. Note that nut-scanner similar function was not modified however.
- FreeBSD DEVD support: generate devd.conf files for USB UPSes This adds a --with-devd-dir=PATH option to ./configure
- The NUT website was moved to a standalone website. A separate code repository and source archive are now available.
- As usual, more bugfixes, cleanup and improvements, on both source code and documentation.

2.6 Release notes for NUT 2.7.1 - what's new since 2.6.5

Note
There was no public NUT 2.7.0 release.

- This release is an interim release, part of the testing series, and the first release after the transition from Subversion to Git. The last release (2.6.5) is almost a year old. A lot of work has been done, but a good amount remains to achieve 2.8.0 goals. Please read the UPGRADING.adoc notes.
- Added support for SSL via the Mozilla NSS library, in addition to the existing OpenSSL support.
- Added a new driver, nutdrv_qx, for Megatec/Qx devices. This driver will eventually replace the blazer_ser and blazer_usb drivers. In particular, it adds support for Voltronic Power devices.
- Increased USB_TIMEOUT to standards-compliant 5.000 seconds in most drivers. This should reduce the number of timeouts on low-speed USB 1.1 devices.
- The jNut Java source has been split into a separate GitHub repository.
- Added many devices to the HCL. Of particular note are many Tripp Lite USB HID PDC models which were tested against NUT by Tripp Lite.
- Reworked some visual elements of the HCL. The output is better tailored for graphical and text-only browsers, but suggestions are welcome for additional accessibility enhancements.
- Also increased timeouts and added redundant commands to improve reliability of mge-utalk driver.
- Added the apcupsd-ups driver to interoperate with apcupsd installations.
- Added documentation on creating subdrivers for snmp-ups and nutdrv_qx.
- Added new drivers for the Riello UPS product line (riello_ser/riello_usb).
- Many improvements to the BCM/XCP drivers have been merged in. This includes an improved data reception loop, and additional mappings.
- Added a few variables to the Powercom HID mappings.
- Updated the apcsmart driver, and renamed the previous driver to apcsmart-old.
- Fixed the battery percentage calculation in the bestfcom driver.
- libnutclient has been added as a C++ alternative to libupsclient.
- Packaging files for Solaris and HP-UX (sponsored by Eaton)
- Fix shutdown of Eaton HID, using usbhid-ups and mge-shut
- usbhid-ups: final fix for APC Back UPS ES. APC Back UPS ES devices have buggy firmware, and this version does not cause a regression. The max_report variable should be set automatically based on the USB identification values.
- nut-scanner: fix crash
- IPMI support can handle more different versions of FreeIPMI
- Support power supplies scan over the network nut-scanner can now scan for power supplies with IPMI over LAN. This is currently limited to IPMI 1.5 only
- Implement a framework to spell check documentation source files, using Aspell. This includes an interactive build target (make spellcheck-interactive), and an automated one (make spellcheck), mainly for QA / Buildbot purpose. Note that a base NUT dictionary is also available (docs/nut.dict), providing a glossary of terms related to power devices and management
- Improve systemd integration
- snmp-ups: Fixed a crash on outlet management, and added delta_ups MIB support. Also fixed mappings for upsBypassVoltage, upsBypassCurrent, and upsBypassPower in three-phase IETF MIB.

2.7 Release notes for NUT 2.6.5 - what's new since 2.6.4

- This release fixes an important regression in upssched: any upssched.conf command that takes a second argument resulted in a defective frame sent to the parent process. Thus, the command was not executed (report and patch from Oliver Schonefeld)
- Website hosting: free NUT from Eaton website hosting

  NUT website (https://www.networkupstools.org) is no longer hosted by Eaton. Arnaud Quette (NUT project leader) has taken over NUT hosting on his own, to give NUT back some independence.

  This effort is also part of a logic to stop crediting Eaton for contributions from others (especially Arnaud Quette, as an individual). The new hosting service is located, as for Arnaud's blog (http://arnaud.quette.fr) on Gandi servers, using PaaS.

  This will allow more flexibility and automation of the release process.
- macosx-ups: new OS X Power Sources meta-driver
  - Mac OS X provides UPS status information in a format similar to what is shown for laptop batteries. This driver will convert that information into a format compatible with NUT (Charles Lepple).
- support for new devices:
  - Eaton ePDU Switched
  - Online Zinto A (USB ID 0x06da:0x0601)
  - REDi Blazer 400VA / 600VA / 800VA
  - UNITEK Alpha650ipF and Alpha650ipE (USB ID 0x0f03:0x0001)
- mge-shut driver has been replaced by a new implementation (newmge-shut). In case of issue with this new version, users can revert to oldmge-shut.
- First NUT virtualization package: NUT now supports integration with VMware ESXI 5.0, through a native VIB package. This is, for the time being, an external effort from René Garcia (refer to the Download section on NUT website). But work is underway to improve this integration, and include it in the NUT source tree
- IPMI support (nut-ipmipsu driver and nut-scanner): prepare for supporting API changes in upcoming FreeIPMI versions 1.1.x and 1.2.x.
- snmp-ups now supports high precision values for APC, and more variables
- the NUT variables and commands namespace has been fixed and completed, with the known and used variables that were missing.
- more bugfixes, cleanup and improvements, on both source code and documentation.

2.8 Release notes for NUT 2.6.4 - what's new since 2.6.3

- This release fixes an important vulnerability in upsd (CVE-2012-2944: upsd can be remotely crashed)

  NUT server (upsd), from versions 2.4.0 to 2.6.3, are exposed to crashes when receiving random data from the network.

  This issue is related to the way NUT parses characters, especially from the network. Non printable characters were missed from strings operation (such as strlen), but still copied to the buffer, causing an overflow.

  Thus, fix NUT parser, to only allow the subset of ASCII charset from `Space` to `~` (Reported by Sebastian Pohle, Alioth bug #313636, CVE-2012-2944)

  A separate patch, which applies to any faulty version, is also available: http://trac.networkupstools.org/projects/nut/changeset/-3633

For more information, refer to the Common Vulnerabilities and Exposures: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2944

- A static source code analysis has been done by Michal Hlavinka from RedHat, using Coverity (12 issues fixed).
- Add new "LIST CLIENTS" and "NETVER" commands to NUT network protocol. "NETVER" allows to retrieve the Network protocol version, while "LIST CLIENTS" provides the list of clients connected to a device. Refer to the developer guide, "Network protocol information" section for more information.
- Support of ranges of values for writable variables has been added, to complete the existing enumerated values mechanism. This will start to appear in some drivers soon, beginning with Eaton. Refer to the developer guide, "Creating a new driver..." section for more information.
- PyNUT.py has been updated to version 1.2.2, adding support for LIST CLIENTS, FSD, HELP and VER (Rene Martín Rodríguez)
- support for new devices:
  - AEG Power Solutions PROTECT HOME
  - more APC SNMP cards
  - ATEK Defensor range
  - all Borri models
  - all COVER ENERGY SA
  - CyberPower OR700LCDRM1U, PR6000LCDRTXL5U and CP1000PFCLCD
  - Dell UPS Network Management Card
  - Dynamix 1000VA USB
  - Eaton Management Card Contact (ref 66104)
  - EVER POWERLINE RT / 11 / 31 and DUO II Pro
  - GE Digital Energy GT Series 1000-3000 VA
  - Gtec models
  - all recent HP serial / USB UPS (G2, G3 and R/T models,) and HP UPS Management Module
  - Ippon INNOVA RT
  - KOLFF BLACK NOVA
  - Lexis X-power Tigra 1kVA
  - Microline C-Lion Innova
  - Online Yunto YQ450
  - PowerShield Defender 1200VA
  - PowerWalker Online VFI LCD, Line-Interactive VI LCD and Line-Interactive VI
  - Riello Netman Plus 102 SNMP Card
  - Tripp-Lite OMNISMART500
- apcsmart has received some fixes to work better on Mac OS X, and in general
- bcmxcp has improved support for battery status, and better supports PW9120 units
- bestfortress has improved Best Fortress LI675VA support
- blazer_ser and blazer_usb now try to automatically estimate high and low voltages, to be able to calculate battery charge; support for online Innova UPS (T, RT and 3/1 T) has been added; Best UPS support has been improved, to prepare for superseding bestups driver
- bestups has also received some care, though users are encouraged to switch to blazer_ser, since bestups will soon be deprecated.

- newmge-shut has been heavily improved. However, replacement of the current mge-shut has been postponed to the next release, due to the CVE issue.
- oneac 0.80 improves support for all families of Oneac (EG, ON, OZ and OB), including more data and instant commands (Bill Elliot).
- usbhid-ups: for Eaton devices, ups.start.auto is now automatically adjusted for shutdown.{return,stayoff} to behave as expected; Liebert firmwares with incorrect exponents have also been addressed.
- snmp-ups now provides support for UPS shutdown, based on usbhid-ups mechanisms (composite commands and fallback). Composite commands are also supported now. This means, for example, that if *shutdown.return* is not supported, a combination of *load.off* + *load.on* may be used; Actual validity of instant commands is now tested before commands addition; Eaton/MGE MIB has been cleaned and completed; 3-phases support has been added to Socomec Netvision MIB; HP/Compaq MIB has been completed, with thresholds, nominal values and more commands.
- nut-scanner now also has libupsclient has a weak runtime dependency; more docs and bugfixes have also happened.
- Provide an Uncomplicated Firewall (UFW) profile (nut.ufw.profile)
- Riello protocols have been officially published in NUT protocols library: https://www.networkupstools.org/ups-protocols.html#_riello
- Duplicate instances of upsd / upsmon are now detected upon startup
- NUT variables namespace has been completed with missing variables and commands that are already known and standard
- upslog now comes with a companion file, for logrotate configuration
- more devices embed NUT for power protection, now including Thecus NAS range
- more bugfixes, cleanup and improvements, on both source code and documentation, with a good bunch from Greg A. Woods.

2.9 Release notes for NUT 2.6.3 - what's new since 2.6.2

- nut-scanner is now more portable, and provides more coherent option names. IPMI support has also been added, to discover local power supplies. This version brings weak runtime dependencies in libnutscan, which allows to compile nut-scanner with all options and to run according to the available dependencies (USB, SNMP, IPMI, ...).
- libnutscan now provides pkg-config support and the needed header files are distributed. Some documentation is also available in the developer guide and manual pages have been updated and completed.
- support for new devices:
  - Cyber Power Systems with SNMP RMCARD (100, 201, 202 and 301)
  - Dynamix 650VA USB
  - LDLC UPS-1200D
  - Tecnoware UPS ERA LCD 0.65
  - Powercom BNT-xxxAP (USB ID 0d9f:0004)
  - Various USB devices using UPSilon 2000 software
- apcsmart has received minor correction.
- bcmxcp_usb now handles disconnection issues and reconnection mechanism.
- blazer_usb enables again inclusion of buggy USB Device and Vendor IDs in udev rules file; language ID support has been added for USB units from LDLC, Dynamix and other no names.
- nut-ipmipsu has also received some improvements.
- snmp-ups has fixed outlets reported current in aphel_genesisII MIB; MGE 3 phases handles better low battery condition; support for Cyber Power Systems with SNMP RMCARD has been added; support of the newer Eaton ePDUs has been improved.

- upsd doesn't anymore fail to start if at least one of the listening interface is available. This is needed on systems where either IPv4 or IPv6 is disabled, and no explicit LISTEN directive has been specified.
- Avahi support is now automatically enabled, upon detection
- jNut (NUT Java interface) adds device discovery support, through a nut-scanner wrapper; jNutWebAPI, a HTTP/JSON web service interface, has also been added to interact with upsd and nut-scanner.
- Base files for HPUX packaging have been added. This is still a work in progress.
- Compilation on IBM AIX has been fixed (namespace conflict with ctypes).
- more bugfixes, cleanup and improvements, on both source code and documentation.

2.10 Release notes for NUT 2.6.2 - what's new since 2.6.1

- NUT can now monitor power supply units (PSU) in servers, through IPMI, using the new experimental *nut-ipmipsu* driver. Users are encouraged to test it, and send feedback and debug output to the development mailing list. This requires GNU FreeIPMI (0.8.5 or higher) development package or files. Thanks goes to Al Chu, FreeIPMI project leader, for his help during this development.
- NUT now provides a tool, called *nut-scanner*, to discover supported devices, both local and remote. nut-scanner will help to ease the configuration step, and power infrastructure discovery.

  This development, sponsored by Eaton, supports the following methods:

  - USB,
  - SNMP,
  - XML/HTTP (from Eaton),
  - NUT servers, using the classic connect or Avahi / mDNS methods.

    IPMI support will be added in the next release.

    A separate library, called *libnutscan*, is also available to provide these feature. Future NUT releases will provides binding for the supported languages (Perl, Python and Java).
- NUT now provides a Java interface called *jNut*. This development, sponsored by Eaton, is currently limited to the client interface. But it will be broaden to device discovery and configuration in the future.

  For more info, refer to nut/scripts/java/README, or the developer guide (chapter *Creating new client*). Javadoc documentation is also provided, along with Java archives (.jar) in the Download section.
- support for new devices:

  - Eaton 3S
  - Cyber Power Systems CP1000AVRLCD
  - various APC models equipped with APC AP9618 management card, including APC Smart-UPS RT XL
  - Orvaldi 750 / 900SP
  - POWEREX VI 1000 LED
  - PowerWalker VI 850 LCD
  - SVEN Power Pro+ series (USB ID ffff:0000)
- A regression has been fixed in udev rules file. This previously caused permission issues to owners of some USB devices.
- Avahi support has been added, for NUT mDNS publication, through a static service file (nut/scripts/avahi/nut.service).

- usbhid-ups has had Eaton completion: some features have been improved, such as *output.voltage.nominal*; 3S Eco control support has been added, along with battery.runtime.low and end of battery life (life cycle monitoring) support; new measurements for 5 PX are also supported now (outlet power factor, power, real power and current).
- apcsmart has been updated to support more variables and features; the previous driver is however still available as *apcsmart-old*, in case of issues.
- bcmxcp now supports per outlet startup and shutdown delays setting; shutdown delay is also used, when available, for outlet.n.shutdown.return instead of the default 3 seconds.
- snmp-ups.c has a new initialization method, that uses sysObjectID, which is a pointer to the preferred MIB of the device, to detect supported devices. This speeds up even more init stage and should render void the use of *mib* option. SNMP v3 session initialization has also been fixed, and Eaton PDU support has been completed.
- Initial support has been added for systemd, the System and Service Manager from RedHat.
- The chapter *NUT configuration management with Augeas* of the developer guide has received some completion: a complete Python Augeas example code is now provided.
- Finally, after years of dedication to NUT, Arjen de Korte is now retired. Sincere thanks to you Arjen from us all.

2.11 Release notes for NUT 2.6.1 - what's new since 2.6.0

- the various recent USB regressions have been definitely fixed.
- NUT now propose a variable to expose UPS efficiency (ups.efficiency). Eaton 5 PX already uses it.
- the Perl module from Gabor Kiss (rewritten from Kit Peters') is now distributed with NUT source code.
- support for new devices:
  - Eaton Ellipse ECO, Powerware 9140, Eaton 5 PX, and ambient sensor on Eaton ePDU managed
  - GE EP series
  - Inform Sinus SS 210
  - IPAR Mini Energy ME 800
  - Mustek Yukai PowerMust 1000 USB
  - Numeric 3000 SW
  - SVEN Power Pro+ series (recent models)
  - Vivaldi EA200 LED
- liebert-esp2: Improved Liebert ESP II support, including UPS shutdown (poweroff), 1 and 3-phase input and output variables, and most input / output / bypass / nominal variables. There is also a fix for the USB to serial cable (Farkas Levente and Spiros Ioannou).
- powercom has improved PowerCom BNT 1500A and BNT-other support, along with driver documentation and code conformance to the NUT rules (Keven L. Ates).
- apcsmart has more improved UPS poweroff support and options (Michal Soltys).
- blazer has also seen some improvements.
- usbhid-ups has completed a bit supported variables for APC and Eaton / MGE.
- on the quality assurance side, Eaton has worked on fixing a few non-conformance issues, like C++ style comments and warnings, using a newly developed verification tool (Prachi Gandhi).
- fix remaining references to LIBSSL_LDFLAGS, instead of LIBSSL_LIBS, which cause unresolved symbol on libupsclient users (Fabrice Coutadeur).
- the website has now a better support for Internet Explorer 6.
- graphic illustrations, used for the Features page on the website Features and chapter of the user manual, have been refreshed (courtesy of Eaton).
- more bugfixes, cleanup and improvements, on both source code and documentation.

2.12 Release notes for NUT 2.6.0 - what's new since 2.4.3

Note
Per original semantic versioning, there were no public NUT 2.5.x releases.

- the main focus of this release is the complete documentation revamping, using AsciiDoc. This includes a new website, user manual, developer guide, packager guide and manual pages, available in various formats (single and multiple pages HTML, and PDF at the moment).

  Be sure to check the `--with-doc` option help of `configure` script, and docs/configure.txt for more information.

- Add Augeas support, to provide easy NUT configuration management, through tools and development APIs. For more information, refer to the developer guide, or scripts/augeas/README.adoc in the source directory.

- support for new devices:
  - APC 5G
  - Eaton PowerWare 5119 RM (smart mode using upscode2 driver)
  - Eaton Best Ferrups (using older ConnectUPS card)
  - Eaton 9395 (serial interface)
  - Eaton ConnectUPS X / BD / E Slot
  - HP T1000 INTL
  - HP T1500 INTL
  - HP T750 G2
  - HP R1500 G2 INTL
  - iDowell iBox UPS
  - Tripp Lite SmartOnline SU1000XLA
  - Tripp Lite Smart1000LCD
  - and some more USB/HID devices IDs
  - CyberPower CP1500AVRLCD and CP1350AVRLCD
  - PowerWalker Line-Interactive VI 1400
  - Rocketfish RF-1000VA / RF-1025VA

- usbhid-ups has better support for shutting down APC SmartUPS RM series, and finally fix the "buffer size" issue, which was breaking some devices data retrieval, or truncating some data on others.

- snmp-ups now support SNMP v3 and its security parameters. IETF MIB support has also been extended.

- fix dummy-ups simulation driver status handling bug, and add the capability to remove exposed variables on the fly.

- the belkin driver now support control commands and status reporting for beeper and battery test.

- the powerpanel driver supports more older CyberPower units.

- mge-utalk, upscode2, blazer and liebert-esp2 have also received some care, and been improved.

- NUT-Monitor and the PyNUT client module have been updated to 1.3, adding more features like automatic connection to the first local device and i18n support.

- improve configure time dependencies checking and processing.

- improve older Unix systems support (HP-UX, AIX, ...) for missing functions.

- refresh and improve USB helper files (udev and UPower).

- more generation automation: the ChangeLog file is now generated automatically at distribution time, along with the files needed for the website hardware compatibility list.
- SSL support has also received some improvements.
- tcp-wrapper now allows hostnames in /etc/hosts.allow too (not only IPv4 and/or IPv6 addresses).
- many bugfixes, cleanup and improvements.

2.13 Release notes for NUT 2.4.3 - what's new since 2.4.2

- this is a bugfix release that only solves the regression on IPv6 activation.

2.14 Release notes for NUT 2.4.2 - what's new since 2.4.1

- the general USB support has been vastly improved, including many bug fixes, better OS support, new features and devices.
- NUT now talks to Solar Controller Devices with the new ivtscd driver.
- the snmp-ups driver supports more PDU, with a smaller disk footprint.
- apcsmart supports more older SmartUPS and Matrix units.
- the bestfortress driver is resurrected.
- the virtual driver has been renamed to *clone*.
- the netxml-ups driver has received some care.
- various debugging and development improvements have been done, around driver output; dummy-ups with more interaction and scripting and the device-recorder.sh script.
- the build system has received many bugfixes and improvements.
- the UPower (previously known as DeviceKit-power) rules file is now generated by NUT.
- support for new devices:
  - Apollo 1000A and 1000F
  - various Baytech RPC
  - old Best Power Fortress
  - Cyber Power Systems PR3000E, CP 1500C and OR2200LCDRM2U
  - all the new Dell UPS range (serial, USB and network)
  - Eaton E Series NV and DX UPS, and Powerware 9130
  - older HP T500 and T750, newer T750 INTL (USB) and R1500 G2 (serial)
  - Inform Informer Compact 1000VA
  - many serial and USB devices from Ippon, like Back Comfo Pro, Smart Power Pro and Smart Winner
  - IVT SCD series
  - Liebert GXT2-3000RT230 and PowerSure PSA
  - Mustek PowerMust 424 / 636 / 848 USB
  - all new PowerCOM USB devices with HID PDC interface
  - Tripp-Lite INTERNETOFFICE700, SMART700USB and ECO550UPS
  - UPSonic DS-800 (USB)

2.15 Release notes for NUT 2.4.1 - what's new since 2.4.0

- the microdowell driver has appeared to support various MicroDowell Enterprise units (see the "new devices" list below).
- support for new devices:
  - MicroDowell Enterprise B8, B10, N8, N11, N15, N20, N22, N30, N40, N50, N60 and HiBox ST.
- NUT-Monitor now better handles the ups.status field, and has switched to version 1.1.
- the situation of the build toolchain has been fixed, with regard to the "make clean" target and the wrongly removed generated USB files. This broke further configure call.

2.16 Release notes for NUT 2.4.0 - what's new since 2.2.2

Note
Per original semantic versioning, there were no public NUT 2.3.x releases.

- preliminary support for Power Distribution Units (PDUs): NUT can now support PDUs, either natively (ie using NUT snmp-ups driver), or through a binding to the Powerman daemon. The list of supported PDUs is already quite long, including:
  - Eaton ePDUs (Managed and Monitored),
  - some Aphel models,
  - some Raritan PDUs,
  - and the whole list of Powerman supported devices: http://powerman.sourceforge.net/supported.html
- support for new devices:
  - the various PDUs cited above
  - Chloride Desk Power 650
  - Cyber Power Systems Value 400E/600E/800E (USB models)
  - Delta GES602N
  - Digitus DN-170020
  - the whole Eaton ranges (mostly composed of MGE Office Protection Systems and Powerware units) including BladeUPS
  - Forza Power Technologies SL-1001
  - HP PowerTrust 2997A
  - HP R/T 2200 G2
  - Infosec XP 1000 and XP 500
  - Ippon Back Power Pro (serial and USB)
  - Kebo 1200D/D Series
  - Liebert PowerSure Personal XT
  - MGE Office Protection Systems Protection Station
  - Neus 400va and 600va
  - Phasak 400VA and 600VA
  - Plexus 500VA
  - Powercom Black Knight PRO / King PRO and Imperial
  - PowerKinetics BlackOut Buster
  - Sweex 1000 USB
  - UNITEK Alpha 500

  - WinPower CPM-800

- NUT now embeds Python client support through the PyNUTClient module and the NUT-Monitor application. Both are from David Goncalves, and are still available from http://www.lestat.st. For more information, refer to scripts/python/README.
- the dummy-ups driver now supports a "repeater" mode. This allows it to act as a NUT client, and to forward data. This can be useful for supervision and load sharing purposes.
- tcp-wrappers support has been added to the upsd server, to grant users access by source IP for commands that require to be logged into the server. This replaces the previous internal implementation (ACL in upsd.conf).
- the nut.conf file has been introduced to standardize startup configuration across the various systems.
- NUT now ships a bash completion function for *upsc* command (scripts/misc/nut.bash_completion). Simply copy it to /etc/bash_comple
- many internal changes to improve maintainability, while lowering the maintenance cost (thus allowing developers to focus on what matters: the code!). Examples of this are:
- the USB information automatic extraction to generate the various USB helper files,
- the upsdrv_info_t structure to track more driver information, and remove the need for the upsdrv_banner() function
- common USB code refactoring, as it is done for the serial functions.
- tons of bugfixes, cleanup and improvements to make NUT stronger than ever!

2.17 Release notes for NUT 2.2.2 - what's new since 2.2.1

- support for new devices: APC BACK-UPS XS LCD, Atlantis Land, Mustek Powermust Office 650, Oneac XAU models, Powerware PW5115 and PW9120 (USB), Nitram Elite 2005
- Integrated Power Management (NUT HAL integration) has reached a major milestone: it is now the most advanced UPS integration into Power Management layer known in existing OSs. It has received many corrections and improvements, and allows to PowerOff the UPS at the end of a power cycle (which is the most important feature, not supported on other systems). The various files are now installed into the correct location.
- the usbhid-ups driver has received attention. Most notably, the shutdown handling has been reworked, and support for MGE UPS SYSTEMS 3 phases units has been added.
- snmp-ups now supports MGE* Environment Sensor (ref 66 846). The ambient.temperature reporting has also been fixed for units other than APC.
- the netxml-ups driver has appeared to support MGE* network HTTP/XML cards.
- NUT now distributes by default the shared version of libupsclient (version 1.0.0), and use this for the provided clients (upsmon, upsc, upsrw, upscmd). This is part of an effort to reduce NUT's footprint, both on disk and in memory.
- powerpanel has reach a new step toward the replacement of nitram and cpsups drivers. The final step is scheduled for NUT 2.4.
- many changes, cleanup and fixes to the NUT core and various drivers.

2.18 Release notes for NUT 2.2.1 - what's new since 2.2.0

- support for new devices:
  - all MGE Office Protection Systems units
  - Advice TopGuard 2000
  - Belkin F6H375-USB
  - Dynamix UPS1700D

  - Effekta RM2000MH,
  - Jageson Technology Jasuny USPS
  - Powercom SMK-1500A and SXL-1500A
  - PowerWalker Line-Interactive VI 400/800 and 600
  - Powerware 9110
  - UNITEK Alpha 2600
  - UPSonic CXR1000
  - some vintage serial APC UPSs
- the usbhid-ups driver has been improved, and fixed in many areas, through a backport of the development (trunk) version.
- the udev rules, for Linux hotplug support of the USB UPSs, has been updated to support kernel newer than 2.6.22.
- the megatec and megatec_usb drivers have also been backported from the development (trunk) version.
- the client development files have also received some care: the upsclient pkg-config file has been fixed, and the upsclient.h file allows older NUT clients to continue using the UPSCONN structure.

2.19 Release notes for NUT 2.2.0 - what's new since 2.0.5

Note
Per original semantic versioning, there were no public NUT 2.1.x releases.

- The new build infrastructure, using automake, is now used. This has major impact on the compilation and installation procedures, and thus on the NUT packaging. For more information, refer to UPGRADING.adoc and packaging/debian/ for an example of migration.
- NUT now provides support for FreeDesktop Hardware Abstraction Layer (HAL) which brings full Plug And Play experience to USB UPS owners. For more information, refer to docs/nut-hal.txt.
- support for new devices:
  - Ablerex 625L
  - ActivePower 400VA, 2000VA;
  - Belkin Home Office F6H350-SER, F6H500-SER, F6H650-SER
  - Belkin Office Series F6C550-AVR
  - Belkin Universal UPS F6C100-UNV (USB), F6C1100-UNV (USB), F6C1200-UNV (USB), F6H350deUNV (serial), F6H350ukUN (serial), F6H650ukUNV (serial)
  - Compaq R3000h
  - Cyber Power Systems PR2200
  - Dynex DX-800U
  - Geek Squad GS1285U
  - Krauler UP-M500VA
  - Mecer ME-2000
  - MGE UPS SYSTEMS Ellipse MAX
  - Online Zinto D
  - PowerTech SMK-800
  - SVEN Power Pro+ series
  - Power Smart RM 2000

  - Tripp-Lite SmartOnline SU1500RTXL2ua, smart2200RMXL2U.

- added IPv6 support,
- the newmge-shut driver has appeared. This one uses the same HID core as usbhid-ups, but communicate over a serial link. It will eventually replace the current mge-shut driver.
- client commands (upsc, upsrw and upscmd): hostname is now optional, and defaults to "localhost"
- many drivers have been improved and have received bug fixes: powerpanel, megatec, megatec_usb, safenet, tripplite_usb, gamatronic,
- the hotplug and udev scripts, in charge of setting the right permissions on the USB devices, are now installed automatically when appropriate.
- more generally, the NUT core and documentation, including the manpages, have been improved and updated.

2.20 Release notes for NUT 2.0.5 - what's new since 2.0.4

This release is a backport of the development version. Many changes have already been backported previously. Thus it is more a synchronization release, though it includes many bugfixes and support for new models.

- support for new devices:
  - APC Smart-UPS with 6TI firmware
  - Belkin Small Enterprise F6C1500-TW-RK
  - Compaq R3000 XR, R5500 XR
  - Cyber Power 550SL, 725SL, 685AVR, 800AVR, 1200AVR, AE550
  - Eltek
  - Inform GUARD
  - Microsol Rhino
  - Opti-UPS PowerES 420E
  - PowerMan RealSmart, BackPro
  - Powerware PW9315 3-phase
  - SOLA 305
  - Tripp-Lite SMART550USB, SMART2200RMXL2U, OMNI1000LCD, OMNI900LCD, OMNI650LCD, 1500 LCD, AVR550U
  - Viewsonic PowerES 420E
- bcmxcp: added 3-phase support
- megatec: better hardware support, more instant commands
- mge-hid: support more instant commands
- newhidups: fixed APC and Tripp Lite bugs, various memory bugs, improved report buffering, improved Solaris support, added *-x explore* option for easy diagnosis of new devices
- solis: shutdown programming, support new cables, Solaris support
- tripplite_usb: updated SMARTPRO support, fixed OL/OB reporting, better error handling, some memory bugs
- new dummy-ups driver simulator
- added HTML interface for access to CGI scripts

2.21 Release notes for NUT 2.0.4 - what's new since 2.0.3

- The newhidups critical bug (segmentation fault) has been fixed. It has also received some more care, like bugfixes and new models support and enhancement for Solaris. [Peter Selinger and Arnaud Quette]
- A bug has been fixed in NUT core to support resuming from suspend-to-disk. This should also fix other similar issues, like time synchronization through the NTP - Network Time Protocol. [Arjen de Korte]
- The mge-shut driver now better detects the Low Battery status, support new models and fixes some wrong status and data. It also fixes some issue where the UPS wasn't restarting (refer to mge-shut manpage). [Arnaud Quette]
- The genericups custom configuration through ups.conf is working again [Arjen de Korte]
- The genericups driver type 22 also support CyberPower 725SL (and maybe others SL models) [David Kaufman]
- The new megatec driver, which will replace a bunch of drivers by nut 2.2 (refer to docs/megatec.txt and UPGRADING.adoc) has been backported from the trunk (Development tree). The powermust driver has also received some attention. [Carlos Rodrigues]
- The new rhino driver was added to support Microsol Rhino UPS hardware The solis has also been improved for Solaris compatibility, and internal / external shutdown programming. solis can now save external shutdown programming to ups, and support new cables for solis 3 [Silvino B. MagalhÃ£es]
- Several fixes and improvements have been made to upsrw, upsset, cpsups, tripplite_usb and the FAQ. [Arjen de Korte and Charles Lepple]

2.22 Release notes for NUT 2.0.3 - what's new since 2.0.2

- The recent and major newhidups changes have been backported from the Development tree. It now:
- supports models from MGE UPS SYSTEMS, APC and Belkin. Mustek and Unitek units are also recognized for development purpose,
- handles better device reopening, after a disconnection,
- handles multiple devices, with several parameters to find the right UPS. [Peter Selinger, Charles Lepple and Arnaud Quette]
- The bcmxcp_usb driver has been added to support Powerware USB units. [Wolfgang Ocker and Kjell Claesson]
- The tripplite_usb driver has been added to support Tripp Lite USB units. [Charles Lepple]
- The sec driver is back as gamatronic [Gamatronic, Nadav Moskovitch]
- The genericups driver has received official care from Gamatronic to add support for the Gamatronic UPS with alarm interface. [Gamatronic, Nadav Moskovitch]
- The powermust driver now supports Soyntec Sekury C 500 and C 800 units. [Hanno Borns]
- The mge-shut driver has received a bit of attention too, and enhance ups.model retrieval for some specific case (release 0.65)
- The drivers don't change to the "statepath" directory anymore at initialization time if called using -k. This avoid unneeded failure to poweroff the UPS if /var is already unmounted. [Gaspar Bakos]
- The belkinunv driver now supports Belkin F6C1100-UNV [Dave Breiland]
- The isbmex driver has been upgraded to version 0.05, which fixes various errors in formulas, add shutdown capability and revert back baudrate to B9600 (instead of B2400), as it broke the communication [Ricardo Martinezgarza]
- The support of Sysgration UPGUARDS Pro650 in fentonups has been fixed [Simon J. Rowe]
- The packaging files for Red Hat have received various fixes [Thomas Jarosch]
- The solis driver has been fixed to avoid a naming collision and compile on Solaris [Paweł Kierdelewicz]
- The snmp-ups driver has corrected the problem when exposing certain time data.

2.23 Release notes for NUT 2.0.2 - what's new since 2.0.1

- the newhidups USB driver has been improved a lot and is no more experimental. It also now has a basic APC support, which will soon replace the legacy hidups driver.
- The mge-utalk driver has improved its support for old units.
- The mge-shut driver has been improved for restart/shutdown sequences which was previously blocking the serial port.
- The general MGE support has been added Pulsar EXtreme C / EX RT, Comet EX RT, Pulsar SV, Pulsar PSX, Ellipse Office and NOVA AVR USB.
- The genericups driver now supports Generic RUPS 2000, AEC MiniGuard UPS 700 (using Megatec M2501 cable), and Powerware 3110. [Nick Barnes, Paul Andreassen]
- The powermust driver now supports SquareOne Power QP1000, Mustek PowerMust 1400VA Plus and 2000VA USB. [Carlos Rodrigues]
- The fentonups driver has been enhanced and now supports Sysgration UPGUARDS Pro650. [Michel Bouissou, Simon J. Rowe]
- The cpsups driver now supports MicroDowell B.Box BP 500/750/1000/1500. [Armin Diehl]
- The snmp-ups driver now supports Socomec SNMP devices (Netvision MIB), and Powerware ConnectUPS SNMP cards. [Thanos Chatziathanassiou, Olli Salvia]
- The bcmxcp driver is back with support for Powerware UPSs. [Tore Ørpetveit, Kjell Claesson]
- The cyberpower driver now supports CyberPower 1000AVR. [Dave Huang]
- The new solis driver supports Microsol units: Solis 1.0, 1.5, 2.0 and 3.0. [Silvino B. Magalhaes]
- The apcsmart driver has fixed APC600 support.
- The etapro driver fixes brokenness due to ser_get_line use [Marek Michalkiewicz]
- The new upscode2 driver supports Fiskars, Compaq and Powerware devices. [Niels Baggesen, Havard Lygre]
- The tripplite driver has fixed a battery charge bug [Cedric Tefft]

2.24 Release notes for NUT 2.0.1 - what's new since 2.0.0

- The bestuferrups driver has been forked into the new bestfcom driver which has better handling of the inverter status alarm messages and more. [Kent Hill]
- Mustek UPS support returns with two drivers which have overlapping coverage: mustek and powermust. [powermust: Carlos Rodrigues, mustek: Martin Hajduch]
- Additional CyberPower Systems hardware is supported with the new cpsups driver. Three recognized models are the CPS1500AVR, CPS1100VA, and OP500TE. [Walt Holman, Brad Sawatzky]
- The genericups driver can now generate staleness warnings in specific cases where the UPS provides a way to test for its presence. See the "CON" setting in ups.conf for more details. [stan / saticed.me.uk]
- Documentation for monitoring a Back-UPS RS 500 on a system without USB ports has been added to the cables directory. [Martin Edlman]
- The everups driver now supports types 73-76 (NET 700/1000/1400/500-DPC) [hunter]
- The new metasys driver supports Meta System models: Line, HF Millennium, HF Top Line, ECO Network, ECO, Ally HF, Megaline [BlaXwan]
- The ippon driver now allows user-defined settings for the delay before switching off, and the delay before powering on. [Yuri Elizarov]

- The victronups driver is now at version 0.1.9, which adds many instant commands: calibration control, battery and front panel tests, and bypass control. [Gert Lynge]
- The tripplite driver has received a major overhaul to bring it up to working condition for the 2.0 tree, including code cleanups, several new variables, commands, and user-definable parameters. See ChangeLog for more. [Nicholas J Kain]
- The mge-utalk driver has been upgraded to version 0.81, which fixes the lack of read-write variables and loss of sync on models which don't support restoring settings. [Arnaud Quette]
- The Micro Ferrups model RE is now supported by the bestuferrups driver. The driver will also now read the ambient temperature and will no longer constantly report the data as stale. [Tim Thompson]
- The fentonups driver's init sequence has been reworked to work better with some hardware, including a fix to the parser code. [MLH]
- A workaround has been added to the hidups driver to avoid variables which are stuck by calling HIDIOCINITREPORT in every poll. [Stuart D. Gathman]
- SOLA 610 UPS hardware and others which do not support the ID command may now be monitored by the bestups driver after forcing ID= in ups.conf. [Jason White]
- "pollinterval" is now available via driver.parameter for consistency. [Arnaud Quette]
- The mge-shut and newhidups drivers, along with the supporting hidparser/libhid code have received many updates, including lowering USB bandwidth consumption, driver unbinding (only in Linux), code cleanups, and more which can be seen in the ChangeLog file. [Arnaud Quette]
- The fentonups driver now recognizes several more Megatec protocol units:
  - SuperPower HP360, Hope-550 [Denis Zaika]
  - Unitek Alpha 1000is [Antoine Cuvellard]
- Some variables like uc_sigmask were renamed to avoid clashes with symbols on systems like HP/UX.
- All man pages have been reworked to switch literal "-" characters to hyphens or "\-" as appropriate. [Shaul Karl]
- upssched's CANCEL events were broken following the change to text-based socket messages in 1.5 and have been fixed. [Steven Schoch]
- Calls to varargs functions with raw strings from the config files without an intervening "%s" have been fixed in upsmon, upssched, snmp-ups and upsd. [Ulf Harnhammar]

2.25 Release notes for NUT 2.0.0 - what's new since 1.4.x

- The new naming scheme for variables and commands (introduced in 1.4) is now mandatory. The 1.4 tree supported both the old (STATUS) and the new (ups.status) as a transitional release, and now that time is over.

  This means that 2.0 is generally smaller than 1.4 code, since the interim compatibility hacks have been removed.
- New serial handling code has been added, with greatly simplified operations. The old mess involving repeated calls to sigaction, alarm, and read has been condensed to a select-read loop.

  This change allows drivers which don't do any serial communications at all (hidups, snmp-ups) to drop that baggage, so they are a bit smaller when compiled.
- The drivers now recognize "chroot=' and *user*= in the global section of ups.conf. This means you don't have to use -r and -u when starting upsdrvctl.
- upsmon now supports the -K argument to check for the presence of the POWERDOWNFLAG file. If it exists and contains the magic string, then upsmon will exit(EXIT_SUCCESS). Otherwise, it will exit(EXIT_FAILURE).

  This feature can be used to simplify shutdown scripts, since now you don't have to keep the script in sync with the upsmon.conf.

- Many small things like signed value comparisons, int vs. size_t and proper use of const/struct were fixed throughout the source. These were mostly for correctness, but a few potential bugs involving very big or very small numbers were fixed at the same time.
- The access control system in upsd.conf has been reworked and simplified. Since access levels have become meaningless in recent releases, the new system is just ACCEPT <acl> or REJECT <acl>.

  If you are upgrading from a previous version of the software, you will have to edit your upsd.conf to use this method. See the UPGRADING.adoc file for more details.
- The build process now halts when make fails in one of the subdirectories. [Petter Reinholdtsen, Charles Lepple]
- Helper data for using upsclient via pkgconfig is now created if pkgconfig is detected when configure runs. [Arnaud Quette]
- The polling interval in drivers may now be set with *pollinterval* in ups.conf. [Gabriel Faber]
- Blazer UPS equipment is now supported with the blazer driver. [Phil Hutton]
- Energizer USB UPS hardware is now supported on Linux with a new experimental driver. [Viktor T. Toth]
- The newhidups driver has been merged as the first step towards portable USB UPS support. This will eventually replace the old Linux-only hidups driver. The newhidups driver is tagged experimental since it is under active development. [Arnaud Quette, Charles Lepple]
- The newapc driver has been renamed to apcsmart, replacing the old driver with that name. If you used the newapc driver, be sure to delete the old binary and fix your ups.conf.
- The apcsmart driver now supports asynchronous notification data from the hardware, which means it can wake up as soon as something happens. This affects the OL/OB/LB/RB data in ups.status, and generally reduces the latency in dispatching status changes by a few seconds.
- The apcsmart driver can now support quirky hardware which does not provide the usual listing of valid command characters. This feature is necessary to monitor new models like the APC CS 350 and old ones like the Matrix 5000. It also now has sdtype=4 to handle the strange shutdown behavior on the CS series.
- The belkin driver now works around broken firmware version 001, avoiding a lengthy delay at startup. It also implements the shutdown sequence differently, and should actually work on more hardware now.
- The bestups driver has been slowed down to play nicer with the hardware, and is much more reliable as a result. Among other things, it should always detect the UPS on the first try, meaning no more "dot dot dot" when it starts.
- The cyberpower driver is no longer tagged experimental, and now supports powering off the load. It also supports battery tests via instcmds.
- Effekta MT 2000 RM hardware is now supported by the fentonups driver. [Christoph Moar]
- The new safenet driver supports UPS hardware that uses the protocol of the same name. This includes models from many manufacturers, including Fairstone, Fenton, Gemini, Powerwell, Repotec, Soltec and Sweex. See the README or driver.list for the full details. [Arjen de Korte]
- The genericups driver now has type 20 to monitor the Powerware 5119 RM. See http://lists.exploits.org/ups/Oct2003/00052.html. [Daniel Thompson]
- The belkinunv driver has been added to allow monitoring Belkin Universal UPS hardware. [Peter Selinger]
- Cyber Power Systems 1100AVR hardware which has a different protocol than the existing binary type (supported by *cyberpower*) is now supported by the experimental cyberpower1100 driver. [Walt Holman]
- upsdrvctl now returns success or failure information in the exit code. Any failure during a requested operation will result in a nonzero value (specifically EXIT_FAILURE).

2.26 Release notes for NUT 1.4.0 - what's new since 1.2.x

- The drivers and upsd now communicate over Unix domain sockets instead of state files, shared memory, or state files with mmap. This change makes many things possible, including the new dynamic variable and command naming scheme described below.

  There is a new development tool called sockdebug in the server directory for debugging driver-server communications on the sockets.

- The old static variable scheme has been replaced by a new dynamic implementation. Vague names have been turned into meaningful names that fit into an organized system. UTILITY is now input.voltage. OUTVOLT is now output.voltage.

  This also applies to the names of instant commands. BTEST1 is test.battery.start, and BTEST0 is test.battery.stop.

  The old names are still supported for compatibility with older clients. This compatibility mode will be maintained throughout the 1.4 series, and will be gone by the release of 2.0. Users with older clients are encouraged to upgrade their software during this time.

- The network protocol has been expanded to handle these new names. Older functions which only apply to the old names will continue to be supported through the 1.4 series.

- The drivers and server (upsd) can now change their user ids and chroot themselves with the new -u and -r arguments. This lets you create a "chroot jail" with the bare minimum components.

  This technique is used to provide a higher degree of security. If someone exploited upsd to get a shell somehow, they would be stuck in the jail.

- upssched now explicitly confirms reception of timer commands before exiting. This was done to avoid a race where one process would exit right when another one was starting. The second one would believe its command had been handled when it had been lost.

- upslog has been reworked to use standard getopt parsing to provide the monitoring settings. The old way of specifying arguments is still supported for backwards compatibility.

  upslog has also been changed to only parse the format string once, rather than doing it every time through the loop. This should provide a minuscule drop in CPU utilization.

- Usernames are now required in upsmon and upsd. This means that you must add a username to your MONITOR lines in upsmon.conf and then create a matching user in upsd.users.

  Installations from the 1.2 era probably already used usernames, so this mostly affects those from 1.0 and before.

- Drivers are now pinged regularly by upsd when they aren't posting updates about the UPS status. This provides another check in the data validation process. If upsd fails to get a response within a few seconds, the UPS will be marked stale.

- A few minor memory leaks were discovered with valgrind and squashed.

- upsstats now reuses connections to upsd when cycling through multiple entries in the hosts.conf. This makes things a bit faster and avoids some of the noise in the syslog.

  This only applies to entries that are adjacent. To take advantage of this feature, you may have to rearrange them per example below.

  Connection reuse for nonadjacent entries may be considered in the future.

```
MONITOR ups-1@host-1 ...
MONITOR ups-1@host-2 ...
MONITOR ups-2@host-2 ...
MONITOR ups-3@host-3 ...
```

- upsd now warns about insecure configuration files at startup. These files (upsd.conf, upsd.users, and the certfile) should only be readable by upsd. Never make them world-readable.
- The programs no longer print "shutting down" when they are just exiting. This was changed to avoid confusion about the term, since "shutting down" has a special meaning in UPS software.
- Signal handlers no longer do any significant amount of work. Some of the programs used to do numerous things in there, raising concerns about reentrancy. They now set flags and allow the main loop to do the actual work from there.
- A bug in upsmon where NOTIFYFLAG settings could be ignored was fixed.
- Group handling has been changed. configure no longer accepts --with-group, and the programs no longer setgid() to a hard-coded value. They now setgid() to the primary group of whatever the user value may be.

  This may be compiled in with --with-user as before, and many programs accept -u to override it at runtime.
- The state path is no longer created during *make install*. Users are now expected to create it themselves. This removes a lot of evil complexity from the build and install sequences.
- upsd no longer implements the DROP access command, as it could confuse the clients by getting them out of sync. DROP is now implemented as DENY, which sends an error message. If you use DROP, you should change it to DENY rather than relying on this compatibility measure.
- The belkin driver no longer reports OFF and OL at the same time.
- The bestups driver no longer sleeps during polls, which makes it more responsive to things like instant commands.
- The cyberpower driver now has much better hardware detection code and no longer freezes at startup under some conditions. It also now supports the shutdown function. Instant commands for shutdowns and battery tests were also added.
- The dummyups testing driver has been removed. The dummycons testing driver can do everything that dummyups once did and much more. dummycons is also now built by default for easier testing.
- The newapc driver has been reworked to take advantage of the new internal driver state functions. Some variables without an obvious purpose were dropped.
- The newapc driver now sends all five bytes when using sdtype 1. Previously it didn't send the entire string, and it didn't work. [Don Lewis]
- The hidups driver has been expanded to allow for setting variables, a shutdown sequence, and more. [Arnaud Quette]
- The mge-utalk driver had trouble establishing communications in some cases due to the RTS line being set. This has been fixed.

  The mge-shut driver has been added to the tree, and has replaced the older mge-ellipse driver. [Arnaud Quette, Philippe Marzouk]
- Outlet-level control has been defined in the variable tree, and will be added to drivers where the hardware supports it. This can be used to shut down some components earlier than others to prolong your runtime on battery.

  This is supported in the mge-shut driver now, and may show up in others before long. [Arnaud Quette]
- KIN-2200AP hardware is now recognized by the powercom driver. This change may also support other KIN-xxxxAP equipment. [Preston A. Elder]
- The 1.1kVA UPS is now supported by the bestuferrups driver. This driver was also changed to allow easy addition of more models in the future. [Bob Apodaca]
- The fentonups driver can now handle devices which implement the "I" detection differently, and now supports the Giant/WELI 500 as a result. [Joon Guillen]
- The serial number of the UPS being monitored can now be specified with serial= in ups.conf in the genericups driver. [Shaul Karl]

- The newapc driver now sends ESC to break out of menus when the initial detection fails. Some new APC models have interactive menus on the serial port, and the driver couldn't handle them before.
- The snmp-ups driver now reports ambient temperature and humidity data for APC equipment. It also now supports the shutdown.reboot and shutdown.reboot.graceful commands. [Dmitry Frolov]
- The list of supported variables and commands in the snmp-ups driver has been expanded. [Arnaud Quette, J.W. Hoogervorst]
- Various drivers now report bypass mode with the BYP status word. [Arnaud Quette]
- Energy Sistem equipment is now supported with the esupssmart driver. [Antonio Trujillo Coronado]
- The Tripp-Lite SU series (SmartOnline) is supported with the new tripplitesu driver. [Allan Hessenflow]
- The HP PowerTrust A2994A is now recognized by the hp driver. [Jan Sporbeck]
- Many drivers were cleaned up to perform basic sanity checks on the status data before using it.
- An explicit cleanup function has been added to the driver core to ensure that all dynamic resources are freed before exiting. This is part of the larger process to check for memory leaks and other bad things. [Arnaud Quette]
- upsd now provides variable descriptions from an auxiliary file. This file is optional, which allows for a smaller memory footprint. It can also be edited for localization or other customizations.
- upsimage and upsstats can now render BATTVOLT data. [Andrew R. Ghali]
- String handling has been cleaned up throughout the tree. Calls to functions like strcpy and strcat were either replaced with other (range-checking) functions or were rewritten to avoid it.
- Many compile-time defaults may now be overridden at runtime. In the environment NUT_CONFPATH and NUT_STATEPATH may be used. upsdrvctl has been changed to execve to pass these along to the drivers. ups.conf now supports driverpath=, and upsd.conf supports DATAPATH. [Bryan Henderson]
- The configure --with-gd switches now actually do something useful when gd has been installed outside the default search directories. [Patrik Schindler]
- The inline keyword is now handled properly on systems which do not support it or have it specified as another name. This was breaking compiles on some systems. [Petter Reinholdtsen]

2.27 Release notes for NUT 1.2.2 - what's new since 1.2.1

- The snmp-ups driver has been upgraded and expanded. It now supports multiple MIBs, meaning it can handle RFC 1628, APCC, and MGE equipment. You can pick the right one with "mibs=" in ups.conf.

  Support for setting variable and instant commands is also available. [Arnaud Quette and Dmitry Frolov]
- The powernet driver has been upgraded. It now supports more variables, has cleaner logging, and may now be considered stable. [Dmitry Frolov]
- The hidups driver now supports physical port IDs. This avoids most of the problems where the hiddev* names can jump around too easily. It will now stay in the same place as long as you keep it plugged into the same physical port. See the ChangeLog file for more details. [David Brownell]
- The hidups driver now also supports the MFR variable on APC Back-UPS ES equipment. [Jonathan A. Davis]
- The sms driver has been updated to version 0.70. [Marcio Gomes]
- The bestups driver now recognizes Best Power Axxium Rackmount equipment. [Ales Casar]
- The liebert driver now uses O_NONBLOCK, and should now work consistently on OpenBSD as a result. [Alex Cichowski]

- The liebert driver also now uses debouncing logic on the status lines. It was possible to get false readings that would start a shutdown or just annoy users with excessive onbatt/online notify messages. The new code forces the status to settle down for 3 polls before accepting the new value.

  This means that very short power events may not be detected. The alternative is having your machine shut down just because it decided to wiggle over to OB LB for a few seconds.

- upsmon has had the disconnect logic fixed so the "communications lost" (COMMBAD) notify will actually go out when the connection fails. [Steve Monett]

- upssched now uses a lock file to prevent a race where two could start at the same time. The second upssched would "win", and the first one would be unreachable. This had the side-effect of not being able to cancel timers on the first one.

  If you use upssched, you must define the LOCKFN directive when upgrading to this version, or it will not work. [Gaspar Bakos]

- The packaging and scripts for Red Hat systems have been updated. [Antonino Albanese]

- upsd is now a bit more lenient about access levels in the *numlogins* check, which is what caused the problem in upsmon (next item).

- upsmon no longer gets stuck in slavesync() when upsd is configured to drop certain queries. This usually happened at the worst possible time: in the middle of a shutdown. [John David Garza]

- The upsclient functions now do more sanity checking on data from upsd so a short read won't return garbage to the callers.

- upsset now works properly with ENUM/VARTYPE values for multiple UPSes on a single upsd. [Dmitry Frolov]

- Various portability fixes for building on SGI were applied. [Andrea Suatoni]

- upsd no longer tries to reference a deleted client structure if the client disconnects at the wrong time. Previously, it tried to use that pointer after the sendback() function had already failed on write and deleted the client. This could cause upsd to segfault depending on what areas were accessed. [Patrik Schindler]

2.28 Release notes for NUT 1.2.1 - what's new since 1.2.0

- The sms driver is back, with support for Microlink Manager III hardware. [Marcio Gomes]

- Fideltronik Ares Series hardware is now supported as genericups type 19. [Tomek Orzechowski and Arkadiusz Mikiewicz]

- The drivers no longer silently drop instant commands or set commands from upsd that happen to get fragmented in transit. [linux@horizon.com]

- The old multilink driver is back with a new name: liebert. It supports Liebert UPStation GXE hardware with the contact-closure cable. This is currently an experimental driver as there is no way to power down the load.

- configure now picks up the right flags for gd automatically if gd 2.0.8 or higher is installed. This greatly simplifies the CGI build process for most users.

- Shutdowns on FreeBSD using the genericups driver should work again. [Petri Riihikallio]

2.29 Historic releases

More ancient history is not covered in detail here.

You can see docs/history.txt for more details.

3 NUT Upgrading Notes

This file lists changes that affect users who installed older versions of this software. When upgrading from an older version, be sure to check this file to see if you need to make changes to your system.

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.

This is a good time to point out that for stricter packaging systems, it may be beneficial to add `--enable-option-checking=fatal` to the `./configure` command line, in order to quickly pick up any other removed option flags.

3.1 Changes from 2.8.0 to 2.8.1

- NUT documentation recipes were revised, so many of the text source files were renamed to `*.adoc` pattern. Newly, a `release-notes.pdf` and HTML equivalents are generated. Packages which deliver documentation may need to update the lists of files to ship. [#1953] Developers may be impacted by new `configure --enable-spellcheck` toggle (should add spelling checks to `make check` by default, if tools are available) to facilitate quicker acceptance of contributions. Packaging systems may now want to explicitly disable it, if it blocks package building (pull requests to update the `docs/nut.dict` are a better and welcome solution). [#2067]

- Several improvements regarding simultaneous support of USB devices that were previously deemed "identical" and so NUT driver instances did not start for all of them:

  - Some more drivers should now use the common USB device matching logic and the 7 `ups.conf` options for that [#1763], and man pages were updated to reflect that [#1766];
  - The `nut-scanner` tool should suggest these options in its generated device configuration [#1790]: hopefully these would now suffice for sufficiently unique combinations;
  - The `nut-scanner` tool should also suggest sanity-check violations as comments in its generated device configuration [#1810], e.g. bogus or duplicate serial number values;
  - The common USB matching logic was updated with an `allow_duplicates` flag (caveat emptor!) which may help monitor several related no-name devices on systems that do not discern "bus" and "device" values (although without knowing reliably which one is which... sometimes it is better than nothing) [#1756].

- Work on NUT for Windows branch led to situation-specific definitions of what in POSIX code was all "file descriptors" (an `int` type). Now such entities are named `TYPE_FD`, `TYPE_FD_SER` or `TYPE_FD_SOCK` with some helper macros to name and determine "invalid" values (closed file, etc.) Some of these changes happened in NUT header files, and at this time it was not investigated whether the set of files delivered for third-party code integration (e.g. C/C++ projects binding with `libnutclient` or `libupsclient) is consistent or requires additional definitions/files. If something gets broken by this, it is a bug to address in future [#1556]

- Further revision of public headers delivered by NUT was done, particularly to address lack of common data types (`size_t`, `ssize_t`, `uint16_t`, `time_t` etc.) in third-party client code that earlier sufficed to only include NUT headers. Sort of regression by NUT 2.8.0 (note those consumers still have to re-declare some numeric variable types used) [#1638]

  - For practical example of NUT consumer adaptation (to cater to both old and new API types) please see https://github.com/-collectd/collectd/pull/4043

- Added support for `make install` of PyNUT module and NUT-Monitor desktop application — such activity was earlier done by packages directly; now the packaging recipes may use NUT source-code facilities and package just symlinks as relevant for each distro separately [#1462, #1504]

- The `upsd.conf` listing of `LISTEN` addresses was previously inverted (the last listed address was applied first), which was counter-intuitive and fixed for this release. If user configurations somehow relied on this order (e.g. to prioritize IPv6 vs IPv4 listeners), configuration changes may be needed. [#2012]
- The `upsd` configured to listen on IPv6 addresses should handle only IPv6 (and not IPv4-mappings like it might have done before) to avoid surprises and insecurity — if user configurations somehow relied on this dual support, configuration changes may be needed to specify both desired IP addresses. Note that the daemon logs will now warn if a host name resolves to several addresses (and will only listen on the first hit, as it did before in such cases). [#2012]
- A definitive behavior for `LISTEN *` directives became specified, to try handling both IPv4 and IPv6 "any" address (subject to `upsd` CLI options to only choose one, and to OS abilities). This use-case may be practically implemented as a single IPv6 socket on systems with enabled and required IPv4-mapped IPv6 address support, or as two separate listening sockets - logged messages to this effect (e.g. inability to listen on IPv4 after opening IPv6) are expected on some platforms. End-users may also want to reconfigure their `upsd.conf` files to remove some now-redundant `LISTEN` lines. [#2012]
- Added support for `make sockdebug` for easier developer access to the tool; also if `configure --with-dev` is in effect, it would now be installed to the configured `libexec` location. A man page was also added. [#1936]
- NUT software-only drivers (dummy-ups, clone, clone-outlet) separated from serial drivers in respective Makefile and configure script options - this may impact packaging decisions on some distributions going forward [#1446]
- GPIO category of drivers was added (`--with-gpio` configure script option) - this may impact packaging decisions on some (currently Linux released 2018+) distributions going forward [#1855]
- An explicit `configure --with-nut-scanner` toggle was added, specifically so that build environments requesting `--with-all` but lacking `libltdl` would abort and require the packager either to install the dependency or explicitly forfeit building the tool (some distro packages missed it quietly in the past) [#1560]
- An `upsdebugx_report_search_paths()` method in NUT common code was added, and exposed in `libnutscan.so` builds in particular - API version for the public library was bumped [#317]
- Some environment variable support was added to NUT programs, primarily aimed at wrappers such as init scripts and service unit definitions, allowing to tweak what (and whether) they write into debug traces, and so "make noise" or "bring invaluable insights" to logs or terminal:
  - A `NUT_DEBUG_LEVEL=NUM` envvar allows to temporarily boost debugging of many daemons (`upsd`, `upsmon`, drivers, `upsdrvctl`, `upssched`) without changes to configuration files or scripted command lines. [#1915]
  - A `NUT_DEBUG_PID` envvar (presence) support was added to add current process ID to tags with debug-level identifiers. This may be useful when many NUT daemons write to the same console or log file, such as in containers/plugins for Home Assistant, storage appliances, etc. [#2118]
  - A `NUT_QUIET_INIT_SSL` envvar (presence or "true" value) prevents `libupsclient` consumers (notoriously `upsc`) from reporting whether they have initialized SSL support. [#1662]
  - A `NUT_QUIET_INIT_UPSNOTIFY` envvar (presence or "true" value) prevents daemons which can notify service management frameworks (such as systemd) about passing their lifecycle milestones, to not report loudly if they could not do so (e.g. running on a system without a framework, or misconfigured so they could not report and the OS would restart the false-positively "unresponsive" service). [#2136]
- `configure` script, reference init-script and packaging templates updated to eradicate `@PIDPATH@/nut` ambiguity in favor of `@ALTPIDPATH@` for the unprivileged processes vs. `@PIDPATH@` for those running as root [#1719]
- The "layman report" of NUT configuration options displayed after the run of `configure` script can now be retained and installed by using the `--enable-keep_nut_report_feature` option; packagers are welcome to make use of this, to better keep track of their deliveries [#1826, #1708]
- Renamed generated nut-common.tmpfiles(.in) ⇒ nut-common-tmpfiles.conf(.in) to install a /usr/lib/systemd-tmpfiles/*.conf pattern [#1755]
  - If earlier NUT v2.8.0 package recipes for your Linux distribution dealt with this file, you may have to adjust its name for newer releases.
  - Several other issues have been fixed related to this file and its content, including #1030, #1037, #1117 and #1712

- Extended Linux systemd support with optional notifications about daemon state (READY, RELOADING, STOPPING) and watchdog keep-alive messages. Note that `WatchdogSec=` values are currently NOT pre-set into systemd unit file templates provided by NUT, this is an exercise for end-users based on sizing of their deployments and performance of monitoring station [#1590, #1777]

- snmp-ups: some subdrivers (addressed using the driver parameter `mibs`) were renamed: `pw` is now `eaton_pw_nm2`, and `pxgx_ups` is `eaton_pxg_ups` [#1715]

- The `tools/gitlog2changelog.py.in` script was revised, in particular to generate the `ChangeLog` file more consistently with different versions of Python interpreter, and without breaking the long file paths in the resulting mark-up text. Due to this, a copy of this file distributed with NUT release archives is expected to considerably differ on first glance from its earlier released versions (not just adding lines for the new release, but changing lines in the older releases too) [#1945, #1955]

3.2 Changes from 2.7.4 to 2.8.0

- Note to distribution packagers: this version hopefully learns from many past mistakes, so many custom patches may be no longer needed. If some remain, please consider making pull requests for upstream NUT codebase to share the fixes consistently across the ecosystem. Also note that some new types of drivers (so package groups with unique dependencies) could have appeared since your packaging was written (e.g. with modbus), as well as new features in systemd integration (`nut-driver@instances` and the `nut-driver-enumerator` to manage their population), as well as updated Python 2 and Python 3 support (again, maybe dictating different package groups) as detailed below.

- Due to changes needed to resolve build warnings, mostly about mismatching data types for some variables, some structure definitions and API signatures of several routines had to be changed for argument types, return types, or both. Primarily this change concerns internal implementation details (may impact update of NUT forks with custom drivers using those), but a few changes also happened in header files installed for builds configured `--with-dev` and so may impact `upsclient` and `nutclient` (C++) consumers. At the very least, binaries for those consumers should be rebuilt to remain stable with NUT 2.8.0 and not mismatch int-type sizes and other arguments.

- libusb-1.0: NUT now defaults to building against libusb-1.0 API version if the configure script finds the development headers, falling back to libusb-0.1 if not. Please report any regressions.

- apcupsd-ups: When monitoring a remote apcupsd server, interpret "SHUTTING DOWN" as a NUT "LB" status. If you were relying on the previous behavior (for instance, in a monitor-only situation), please adjust your upsmon settings. Reference: https://github.com/networkupstools/nut/issues/460

- Packagers: the AsciiDoc detection has been reworked to allow NUT to be built from source without requiring asciidoc/a2x (using pre-built man pages from the distribution tarball, for instance). Please double-check that we did not break anything (see docs/configure.txt for options).

- Driver core: options added for standalone mode (scanning for devices without requiring ups.conf) - see docs/man/nutupsdrv.txt for details.

- oldmge-shut has been removed, and replaced by mge-shut.

- New drivers for devices with "Qx" (also known as "Megatec Q*") family of protocols should be developed as sub-drivers in the `nutdrv_qx` framework for USB and Serial connected devices, not as updates/clones of older e.g. `blazer` family and `bestups`. Sources, man pages and start-up messages of such older drivers were marked with "OBSOLETION WARNING".

- liebert-esp2: some multi-phase variable names have been updated to match the rest of NUT.

- netxml-ups: if you have old firmware, or were relying on values being off by a factor of 10, consider the `do_convert_deci` flag. See docs/man/netxml-ups.txt for details.

- snmp-ups: detection of Net-SNMP has been updated to use `pkg-config` by default (if present), rather than `net-snmp-config(-` script(s) as the only option available previously. The scripts tend to specify a lot of options (sometimes platform-specific) in suggested `CFLAGS` and `LIBS` compared to the packaged `pkg-config` information which also works and is more portable. If this change bites your distribution, please bring it up in https://github.com/networkupstools/nut/issues or better yet, post a PR. Also note that `./configure --with-netsnmp-config(=yes)` should set up the preference of the detected script over `pkg-config` information, if both are available, and `--with-netsnmp-config=/path/name` would as well.

- snmp-ups: bit mask values for flags in earlier codebase were defined in a way that caused logically different items to have same numeric values. This was fixed to surely use different definitions (so changing numbers behind some of those macro symbols), and testing with UPS, ePDU and ATS hardware which was available did not expose any practical differences.
- usbhid-ups: numeric data conversion from wire protocol to CPU representation in GetValue() was completely reworked, aiming to be correct on all CPU types. That said, regressions are possible and feedback is welcome.
- nut-scanner: Packagers, take note of the changes to the library search code in common/common.c. Please file an issue if this does not work with your platform.
- dummy-ups can now specify `mode` as a driver argument, and separates the notion of `dummy-once` (new default for `\*.dev` files that do not change) vs. `dummy-loop` (legacy default for `*.seq` and others) [issue #1385]
  - Note this can break third-party test scripts which expected `*.dev` files to work as a looping sequence with a `TIMER` keywords to change values slowly; now such files should get processed to the end once. Specify `mode=dummy-loop` driver option or rename the data file used in the `port` option for legacy behavior. Use/Test-cases which modified such files content externally should not be impacted.
- Python: scripts have been updated to work with Python 3 as well as 2.
  - PyNUT module (protocol binding) supports both Python generations.
  - NUT-Monitor (desktop UI client) got separated into two projects: one with support for Python2 and GTK2, and another for Python3 and Qt5. On operating systems that serve both environments, either of these implementation should be usable. For distributions that deprecated and removed Python2 support, it is a point to consider in NUT packages and their build-time and installation dependencies. The historic filenames for desktop integration (`NUT-Monitor` script and `nut-monitor.desktop`) are still delivered, but now cover a wrapper script which detects the environment capabilities and launches the best suitable UI implementation (if both are available).
- apcsmart: updates to CS "hack" (see docs/man/apcsmart.txt for details)
- upsdebugx(): added `[D#]` prefix to log entries with level > 0 so if any scripts or other tools relied on parsing those messages making some assumptions, they should be updated
- upsdebugx() and related methods are now macros, optionally calling similarly named implementations like s_upsdebugx() as a slight optimization; this may show up in linking of binaries for some customized build scenarios
- libraries, tools and protocol now support a `TRACKING` ID to be used with an `INSTCMD` or `SET VAR` requests; for details see docs/net-protocol.txt and docs/sock-protocol.txt
- upsrw: display the variable type beside ENUM / RANGE
- Augeas: new `--with-augeas-lenses-dir` configure option.

3.3 Changes from 2.7.3 to 2.7.4

- scripts/systemd/nut-server.service.in: Restore systemd relationship since it was preventing upsd from starting whenever one or more drivers, among several, was failing to start
- Fix UPower device matching for recent kernels, since hiddev* devices now have class "usbmisc", rather than "usb"
- macosx-ups: the "port" driver option no longer has any effect
- Network protocol information: default to type NUMBER for variables that are not flagged as STRING . This point is subject to improvements or change in the next release 2.7.5. Refer to docs/net-protocol.txt for more information

3.4 Changes from 2.7.2 to 2.7.3

- The nutdrv_qx(8) driver will eventually supersede bestups(8). It has been tested on a U-series Patriot Pro II. Please test the new driver on your hardware during your next maintenance window, and report any bugs.
- If you are upgrading from a new install of 2.7.1 or 2.7.2, double-check the value of POWERDOWNFLAG in $prefix/etc/upsmon.conf - it has been restored to /etc/killpower as in 2.6.5 and earlier.
- If you use upslog with a large sleep value, you may be interested in adding `killall -SIGUSR1 upslog` to any OB/OL script actions. This will force upslog to write a log entry to catch short power transients.
- Be sure that your SSL keys are readable by the NUT system user. The SSL subsystem is now initialized after `upsd` forks, to work around issues in the NSS library.
- The systemd nut-server.service does not Require nut-driver to be started successfully. This was previously preventing upsd startup, even for just one driver failure among many. This also matches the behavior of sysV initscripts.

3.5 Changes from 2.7.1 to 2.7.2

- upsdrvctl is now installed to $prefix/sbin rather than $driverexec. This usually means moving from /bin to /sbin, apart from few exceptions. In all cases, please adapt your scripts.
- FreeDesktop Hardware Abstraction Layer (HAL) support was removed. Please adapt your packaging files, if you used to distribute the nut-hal-drivers package.
- This is a good time to point out that for stricter packaging systems, it may be beneficial to add "--enable-option-checking=fatal" to the ./configure command line, in order to quickly pick up any other removed option flags.

3.6 Changes from 2.6.5 to 2.7.1

- The apcsmart(8) driver has been replaced by a new implementation. There is a new parameter, *ttymode*, which may help if you have a non-standard serial port, or Windows. In case of issues with this new version, users can revert to apcsmart-old.
- The nutdrv_qx(8) driver will eventually supersede blazer_ser and blazer_usb. Options are not exactly the same, but are documented in the nutdrv_qx man page.
- Mozilla NSS support has been added. The OpenSSL configuration options should be unchanged, but please refer to the upsd.conf(5) and upsmon.conf(5) documentation in case we missed something.
- upsrw(8) now prints out the maximum size of variables. Hopefully you are not parsing the output of upsrw - it would be easier to use one of the NUT libraries, or implement the network protocol yourself.
- The jNut source is now here: https://github.com/networkupstools/jNut

3.7 Changes from 2.6.4 to 2.6.5

- users are encouraged to update to NUT 2.6.5, to fix a regression in upssched.
- mge-shut driver has been replaced by a new implementation (newmge-shut). In case of issue with this new version, users can revert to oldmge-shut.

3.8 Changes from 2.6.3 to 2.6.4

- users are encouraged to update to NUT 2.6.4, to fix upsd vulnerability (CVE-2012-2944: upsd can be remotely crashed).
- users of the bestups driver are encouraged to switch to blazer_ser, since bestups will soon be deprecated.

3.9 Changes from 2.6.2 to 2.6.3

- nothing that affects upgraded systems.

3.10 Changes from 2.6.1 to 2.6.2

- apcsmart driver has been replaced by a new implementation. In case of issue with this new version, users can revert to apcsmart-old.

3.11 Changes from 2.6.0 to 2.6.1

- nothing that affects upgraded systems.

3.12 Changes from 2.4.3 to 2.6.0

- users of the megatec and megatec_usb drivers must respectively switch to blazer_ser and blazer_usb.
- users of the liebertgxt2 driver are advised that the driver name has changed to liebert-esp2.

3.13 Changes from 2.4.2 to 2.4.3

- nothing that affects upgraded systems.

3.14 Changes from 2.4.1 to 2.4.2

- The default subdriver for the blazer_usb driver USB id 06da:0003 has changed. If you use such a device and it is no longer working with this driver, override the *subdriver* default in *ups.conf* (see man 8 blazer).
- NUT ACL and the allowfrom mechanism has been replaced in 2.4.0 by the LISTEN directive and tcp-wrappers respectively. This information was missing below, so a double note has been added.

3.15 Changes from 2.4.0 to 2.4.1

- nothing that affects upgraded systems.

3.16 Changes from 2.2.2 to 2.4.0

- The nut.conf file has been introduced to standardize startup configuration across the various systems.
- The cpsups and nitram drivers have been replaced by the powerpanel driver, and removed from the tree. The cyberpower driver may suffer the same in the future.
- The al175 and energizerups drivers have been removed from the tree, since these were tagged broken for a long time.
- Developers of external client application using libupsclient must rename their "UPSCONN" client structure to "UPSCONN_t".
- The upsd server will now disconnect clients that remain silent for more than 60 seconds.
- The files under scripts/python/client are distributed under GPL 3+, whereas the rest of the files are distributed under GPL 2+. Refer to COPYING for more information.
- The generated udev rules file has been renamed with dash only, no underscore anymore (ie 52-nut-usbups.rules instead of 52_nut-usbups.rules)

3.17 Changes from 2.2.1 to 2.2.2

- The configure option "--with-lib" has been replaced by "--with-dev". This enable the additional build and distribution of the static version of libupsclient, along with the pkg-config helper and manual pages. The default configure option is to distribute only the shared version of libupsclient. This can be overridden by using the "--disable-shared" configure option (distribute static only binaries).
- The UPS poweroff handling of the usbhid-ups driver has been reworked. Though regression is not expected, users of this driver are encouraged to test this feature by calling "upsmon -c fsd" and report any issue on the NUT mailing lists.

3.18 Changes from 2.2.0 to 2.2.1

- nothing that affects upgraded systems. (The below message is repeated due to previous omission)
- Developers of external client application using libupsclient are encouraged to rename their "UPSCONN" client structure to "UPSCONN_t" since the former will disappear by the release of NUT 2.4.

3.19 Changes from 2.0.5 to 2.2.0

- users of the newhidups driver are advised that the driver name has changed to usbhid-ups.
- users of the hidups driver must switch to usbhid-ups.
- users of the following drivers (powermust, blazer, fentonups, mustek, esupssmart, ippon, sms) must switch to megatec, which replaces all these drivers. Please refer to doc/megatec.txt for details.
- users of the mge-shut driver are encouraged to test newmge-shut, which is an alternate driver scheduled to replace mge-shut,
- users of the cpsups driver are encouraged to switch to powerpanel which is scheduled to replace cpsups,
- packagers will have to rework the whole nut packaging due to the major changes in the build system (completely modified, and now using automake). Refer to packaging/debian/ for an example of migration.
- specifying *-a <id>* is now mandatory when starting a driver manually, ie not using upsdrvctl.
- Developers of external client application using libupsclient are encouraged to rename the "UPSCONN" client structure to "UPSCONN_t" since the former will disappear by the release of NUT 2.4.

3.20 Changes from 2.0.4 to 2.0.5

- users of the newhidups driver: the driver is now more strict about refusing to connect to unknown devices. If your device was previously supported, but fails to be recognized now, add *productid=XXXX* to ups.conf. Please report the device to the NUT developer's mailing list.

3.21 Changes from 2.0.3 to 2.0.4

- nothing that affects upgraded systems.
- users of the following drivers (powermust, blazer, fentonups, mustek, esupssmart, ippon, sms, masterguard) are encouraged to switch to megatec, which should replace all these drivers by nut 2.2. For more information, please refer to doc/megatec.txt

3.22 Changes from 2.0.2 to 2.0.3

- nothing that affects upgraded systems.
- hidups users are encouraged to switch to newhidups, as hidups will be removed by nut 2.2.

3.23 Changes from 2.0.1 to 2.0.2

- The newhidups driver, which is the long run USB support approach, needs hotplug files installed to setup the right permissions on device file to operate. Check newhidups manual page for more information.

3.24 Changes from 2.0.0 to 2.0.1

- The cyberpower1100 driver is now called cpsups since it supports more than just one model. If you use this driver, be sure to remove the old binary and update your ups.conf *driver=* setting with the new name.
- The upsstats.html template page has been changed slightly to reflect better HTML compliance, so you may want to update your installed copy accordingly. If you've customized your file, don't just copy the new one over it, or your changes will be lost!

3.25 Changes from 1.4.0 to 2.0.0

- The sample config files are no longer installed by default. If you want to install them, use *make install-conf* for the main programs, and *make install-cgi-conf* for the CGI programs.
- ACCESS is no longer supported in upsd.conf. Use ACCEPT and REJECT. Old way:

```
ACCESS grant all adminbox
ACCESS grant all webserver
ACCESS deny all all
```

New way:

```
ACCEPT adminbox
ACCEPT webserver
REJECT all
```

Note that ACCEPT and REJECT can take multiple arguments, so this will also work:

```
ACCEPT adminbox webserver
REJECT all
```

- The drivers no longer support sddelay in ups.conf or -d on the command line. If you need a delay after calling *upsdrvctl shutdown*, add a call to sleep in your shutdown script.
- The templates used by upsstats have changed considerably to reflect the new variable names. If you use upsstats, you will need to install new copies or edit your existing files to use the new names.
- Nobody needed UDP mode, so it has been removed. The only users seemed to be a few people like me with ancient asapm-ups binaries. If you really want to run asapm-ups again, bug me for the new patch which makes it work with upsclient.
- *make install-misc* is now *make install-lib*. The misc directory has been gone for a long time, and the target was ambiguous.
- The newapc driver has been renamed to apcsmart. If you previously used newapc, make sure you delete the old binary and fix your ups.conf. Otherwise, you may run the old driver from 1.4.

3.26 File trimmed here on changes from 1.2.2 to 1.4.0

For information before this point, start with version 2.4.1 and work back.

4 Project history

This page is an attempt to document how everything came together.

The Network UPS Tools team would like to warmly thank Russell Kroll.

Russell initially started this project, maintaining and improving it for over 8 years (1996 — mid 2005).

4.1 Prototypes and experiments

4.1.1 May 1996: early status hacks

APC's Powerchute was running on kadets.d20.co.edu (a BSD/OS box) with SCO binary emulation. Early test versions ran in cron, pulled status from the log files and wrote them to a .plan file. You could see the results by fingering `pwrchute@kadets.d20.co` while it lasted:

```
Last login Sat May 11 21:33 (MDT) on ttyp0 from intrepid.rmi.net
Plan:
Welcome to the UPS monitor service at kadets.d20.co.edu.
The Smart-UPS attached to kadets generated a report at 14:24:01 on 05/17/96.
During the measured period, the following data points were taken:
Voltage ranged from 115.0 VAC to 116.3 VAC.
The UPS generated 116.3 VAC at 60.00 Hz.
The battery level was at 27.60 volts.
The load placed on the UPS was 024.9 percent.
UPS temperature was measured at 045.0 degrees Celsius.
Measurements are taken every 10 minutes by the upsd daemon.
This report is generated by a script written by Russell Kroll<rkroll@kadets>.
Modified for compatibility with the BSD/OS cron daemon by Neil Schroeder
```

This same status data could also be seen with a web browser, since we had rigged up a CGI wrapper script which called finger.

4.1.2 January 1997: initial protocol tests

Initial tests with a freestanding non-daemon program provided a few basic status registers from the UPS. The 940-0024C cable was not yet understood, so this happened over the [attachment:apcevilhack.jpg evil two-wire serial hack].

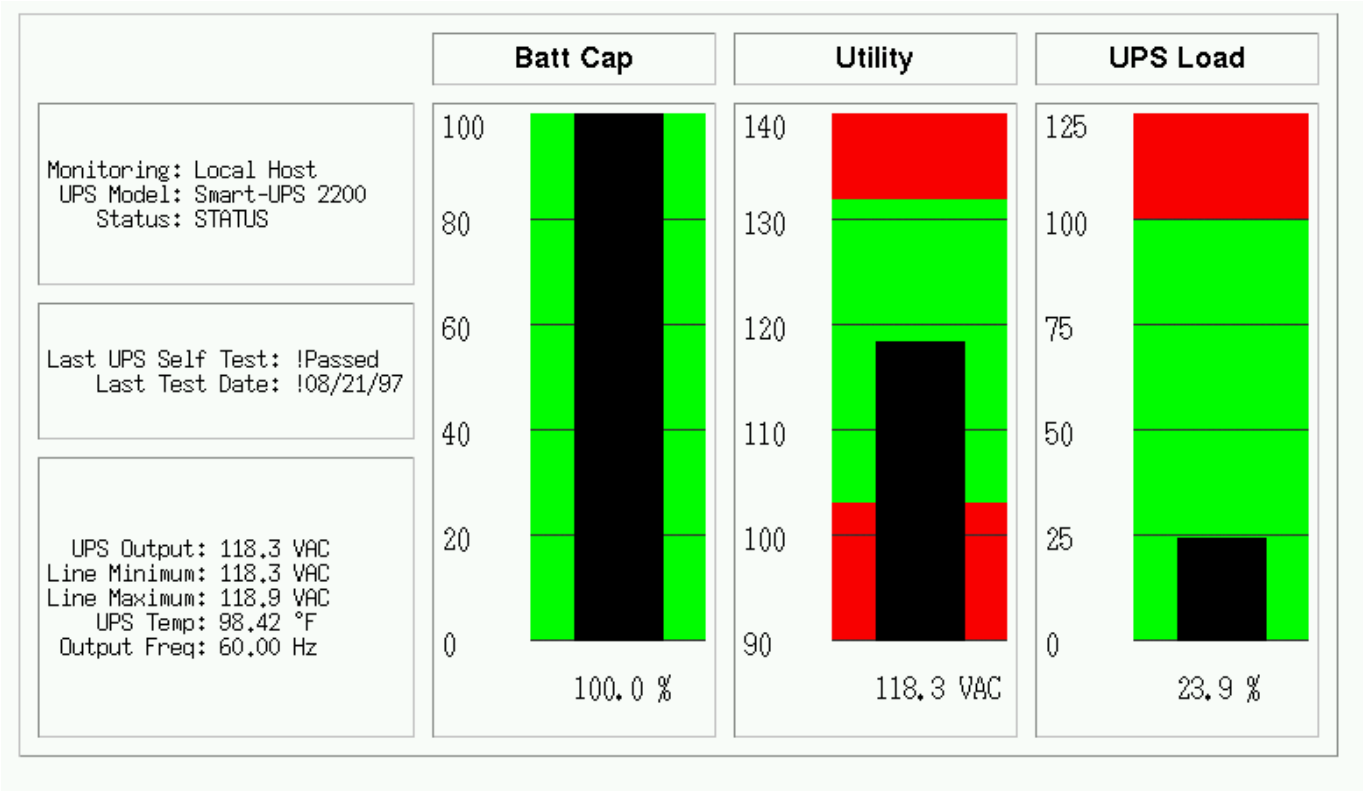
```
Communicating with SMART-UPS 700 S/N WS9643050926 [10/17/96]
Input voltage range: 117.6 VAC - 118.9 VAC
Load is 010.9% of capacity, battery is charged to 100.0% of capacity
```

Note that today's apcsmart driver still displays the serial number when it starts, since it is derived from this original code.

4.1.3 September 1997: first client/server code

The first split daemon/client code was written. upsd spoke directly to the UPS (APC Smart models only) and communicated with upsc by sending binary structures in UDP datagrams.

The first CGI interface existed, but it was all implemented with shell scripts. The main script would call upsc to retrieve status values. Then it would cat a template file through sed to plug them into the page.



upsstats actually has since returned to using templates, despite having a period in the middle when it used hardcoded HTML.

The images were also created with shell scripts. Each script would call upsc to get the right value (utility, upsload, battcap). It then took the value, plugged it into a command file with sed, and passed that into *fly*, a program which used an interpreted language to create images. fly actually uses gd, just like upsimage does today.

This code later evolved into Smart UPS Tools 0.10.

4.2 Smart UPS Tools

4.2.1 March 1998: first public release

Version 0.10 was released on March 10, 1998. It used the same design as the pre-release prototype. This made expansion difficult as the binary structure used for network communications would break any time a new variable was added. Due to byte-ordering and struct alignment issues, the code usually couldn't talk over the network to a system with a different architecture. It was also hopelessly bound to one type of UPS hardware.

Five more releases followed with this design followed. The last was 0.34, released October 27, 1998.

4.2.2 June 1999: Redesigned, rewritten

Following a long period of inactivity and two months of prerelease testing versions, 0.40.0 was released on June 5, 1999. It featured a complete redesign and rewrite of all of the code. The layering was now in three pieces, with the single driver (smartups) separate from the server (upsd).

Clients remained separate as before and still used UDP to talk to the server, but they now used a text-based protocol instead of the brittle binary structs. A typical request like "REQ UTILITY" would be answered with "ANS UTILITY 120.0".

The ups-trust425-625 driver appeared shortly after the release of 0.40.0, marking the first expansion beyond APC hardware.

Over the months that followed, the backupspro driver would be forked from the smartups driver to handle the APC Back-UPS Pro line. Then the backups driver was written to handle the APC Back-UPS contact-closure models. These drivers would later be renamed and recombined, with smartups and backupspro becoming apcsmart, and backups became genericups.

The drivers stored status data in an array. At first, they passed this data to upsd by saving it to a file. upsd would reread this file every few seconds to keep a copy for itself. This was later expanded to allow shared memory mode, where only a stub would remain on the disk. The drivers and server then passed data through the shared memory space.

upsd picked up the ability to monitor multiple drivers on the system, and the "upsname@hostname" scheme was born. Access controls were added, and then the network code was expanded to allow TCP communications, which at this point were on port 3305.

4.3 Network UPS Tools

4.3.1 September 1999: new name, new URL

Several visitors to the web page and subscribers to the mailing lists provided suggestions to rename the project. The old name no longer accurately described it, and it was perilously close to APC's "Smart-UPS" trademark. Rather than risk problems in the future, the name was changed. Kern Sibbald provided the winner: Network UPS Tools, which captures the essence of the project and makes for great short tarball filenames: nut-x.y.z.tar.gz.

The new name was first applied to 0.42.0, released October 31, 1999. This is also when the web pages moved from the old `http://www.exploits.org/~rkroll/smartupstools/` URL to the replacement at `http://www.exploits.org/nu` to coincide with the name change.

More drivers were written and the hardware support continued to grow. upsmon picked up the concepts of what is now known as "primary" and "secondary", and could now handle environments where multiple systems get power from a single UPS.

Manager mode was added to allow changing the value of read/write variables in certain UPS models.

4.3.2 June 2001: common driver core

Up to this point, all of the drivers compiled into freestanding programs, each providing their own implementation of main(). This meant they all had to check the incoming arguments and act uniformly. Unfortunately, not all of the programs behaved the same way, and it was hard to document and use consistently. It also meant that startup scripts had to be edited depending on what kind of hardware was attached.

Starting in 0.45.0, released June 11, 2001, there was a new common core for all drivers called `main.c`. It provided the main function and called back to the `upsdrv_*` functions provided by the hardware-specific part of the drivers. This allowed driver authors to focus on the UPS hardware without worrying about the housekeeping stuff that needs to happen.

This new design provided an obvious way to configure drivers from one file, and so `ups.conf` was born. This eventually spawned upsdrvctl, and now all drivers based on this common core could be started or stopped with one command. Startup scripts now could contain "upsdrvctl start", and it didn't matter what kind of hardware or how many UPSes you had on one system.

Interestingly, at the end of this month, Arnaud Quette entered the UPS world, as a subcontractor of the now defunct MGE UPS SYSTEMS. This marked the start of a future successful collaboration.

4.3.3 May 2002: casting off old drivers, IANA port, towards 1.0

During the 0.45.x series, both the old standalone drivers and the ones which had been converted to the common core were released together. Before the release of 0.50.0 on May 24, 2002, all of the old drivers were removed. While this shrank the list of supported hardware, it set the precedent for removing code which isn't receiving regular maintenance. The assumption is that the code will be brought back up to date by someone if they actually need it. Otherwise, it's just dead weight in the tree.

This change meant that all remaining drivers could be controlled with the `upsdrvctl` and `ups.conf`, allowing the documentation to be greatly simplified. There was no longer any reason to say "do this, unless you have this driver, then do this".

IANA granted an official port number to the project, and the network code switched to port 3493. It had previously been on 3305 which is assigned to `odette-ftp`. 3305 was probably picked in 1997 because it was the fifth project to spawn from some common UDP server code.

After 0.50.1, the 0.99 tree was created to provide a tree which would receive nothing but bug fixes in preparation for the release of 1.0. As it turned out, very few things required fixing, and there were only three releases in this tree.

4.4 Leaving 0.x territory

4.4.1 August 2002: first stable tree: NUT 1.0.0

After nearly 5 years of having a 0.x version number, 1.0.0 was released on August 19, 2002. This milestone meant that all of the base features that you would expect to find were intact: good hardware support, a network server with security controls, and system shutdowns that worked.

The design was showing signs of wear from the rapid expansion, but this was intentionally ignored for the moment. The focus was on getting a good version out that would provide a reasonable base while the design issues could be addressed in the future, and I'm confident that we succeeded.

4.4.2 November 2002: second stable tree: NUT 1.2.0

One day after the release of 1.0.0, 1.1.0 started the new development tree. During that development cycle, the CGI programs were rewritten to use template files instead of hard-coded HTML, thus bringing back the flexibility of the original unreleased prototype from 5 years before. The `multimon` was removed from the tree, as the new `upsstats` could do both jobs by loading different templates.

A new client library called upsclient was created, and it replaced upsfetch. This new library only supported TCP connections, and used an opaque context struct to keep state for each connection. As a result, client programs could now do things that used multiple connections without any conflicts. This was done primarily to allow OpenSSL support, but there were other benefits from the redesign.

upsd and the clients could now use OpenSSL for basic authentication and encryption, but this was not included by default. This was provided as a bonus feature for those users who cared to read about it and enable the option, as the initial setup was complex.

After the 1.1 tree was frozen and deemed complete, it became the second stable tree with the release of 1.2.0 on November 5, 2002.

4.4.3 April 2003: new naming scheme, better driver glue, and an overhauled protocol

Following an extended period with no development tree, 1.3.0 got things moving again on April 13, 2003. The focus of this tree was to rewrite the driver-server communication layer and replace the static naming scheme for variables and commands.

Up to this point, all variables had names like STATUS, UTILITY, and OUTVOLT. They had been created as drivers were added to the tree, and there was little consistency. For example, it probably should have been INVOLT and OUTVOLT, but there was no OUTVOLT originally, so UTILITY was all we had. This same pattern repeated with ACFREQ — is it incoming or outgoing? — and many more.

To solve this problem, all variables and commands were renamed to a hierarchical scheme that had obvious grouping. STATUS became ups.status. UTILITY turned into input.voltage, and OUTVOLT is output.voltage. ACFREQ is input.frequency, and the new output.frequency is also now supported. Every other variable or command was renamed in this fashion.

These variables had been shared between the drivers and upsd as values. That is, for each name like STATUS, there was a #define somewhere in the tree with an INFO_ prefix that gave it a number. INFO_STATUS was 0x0006, INFO_UTILITY was 0x0004, and so on, with each name having a matching number. This number was stored in an int within a structure which was part of the array that was either written to disk or shared memory.

That structure had several restrictions on expansion and was dropped as the data sharing method between the drivers and the server. It was replaced by a new system of text-based messages over Unix domain sockets. Drivers now accepted a short list of commands from upsd, and would push out updates asynchronously. upsd no longer had to poll the state files or shared memory. It could just select all of the driver and client fds and act on events.

At the same time, the network protocol on port 3493 was overhauled to take advantage of the new naming scheme. The existing "REQ STATUS@su700", "ANS STATUS@su700 OL" scheme was showing signs of age, and it really only supported the UPS name (@su700) as an afterthought. The new protocol would now use commands like GET and LIST, leading to exchanges like "GET VAR su700 ups.status" and "VAR su700 ups.status OL". These responses contain enough data to stand alone, so clients can now handle them asynchronously.

4.4.4 July 2003: third stable tree: NUT 1.4.0

On July 25, 2003, 1.4.0 was released. It contained support for both the old "REQ" style protocol (with names like STATUS), and the new "GET" style protocol (with names like ups.status). This tree is provided to bridge the gap between all of the old releases and the upcoming 2.0.

2.0 will be released without support for the old REQ/STATUS protocol. The hope is that client authors and those who have implemented their own monitoring software will use the 1.4 cycle to change to the new protocol. The 1.4 releases contain a lot of compatibility code to make sure both work at the same time.

4.4.5 July 2003: pushing towards 2.0

1.5.0 forked from 1.4.0 and was released on July 29, 2003. The first changes were to throw out anything which was providing compatibility with the older versions of the software. This means that 1.5 and the eventual 2.0 will not talk to anything older than 1.4.

This tree continues to evolve with new serial routines for the drivers which are intended to replace the aging upscommon code which dates back to the early 0.x releases. The original routines would call alarm and read in a tight loop while fetching characters. The new functions are much cleaner, and wait for data with select. This makes for much cleaner code and easier strace/ktrace logs, since the number of syscalls has been greatly reduced.

There has also been a push to make sure the data from the UPS is well-formed and is actually usable before sending updates out to upsd. This started during 1.3 as drivers were adapted to use the dstate functions and the new variable/command names. Some drivers which were not converted to the new naming scheme or didn't do sanity checks on the incoming UPS data from the serial port were dropped from the tree.

This tree was released as 2.0.0.

4.5 Backwards and Forwards Compatibility (NUT v1.x vs. v2.x)

The old network code spans a range from about 0.41.1 when TCP support was introduced up to the recent 1.4 series. It used variable names like STATUS, UTILITY, and LOADPCT. Many of these names go back to the earliest prototypes of this software from 1997. At that point there was no way to know that so many drivers would come along and introduce so many new variables and commands. The resulting mess grew out of control over the years.

During the 1.3 development cycle, all variables and instant commands were renamed to fit into a tree-like structure. There are major groups, like input, output and battery. Members of those groups have been arranged to make sense - input.voltage and output.voltage compliment each other. The old names were UTILITY and OUTVOLT. The benefits in this change are obvious.

The 1.4 clients can talk to either type of server, and can handle either naming scheme. 1.4 servers have a compatibility mode where they can answer queries for both names, even though the drivers are internally using the new format.

When 1.4 clients talk to 1.4 or 2.0 (or more recent) servers, they will use the new names.

Here's a table to make it easier to visualize:

	Server version			
Client version	1.0	1.2	1.4	2.0+
1.0	yes	yes	yes	no
1.2	yes	yes	yes	no
1.4	yes	yes	yes	yes
2.0+	no	no	yes	yes

Version 2.0, and more recent, do not contain backwards compatibility for the old protocol and variable/command names. As a result, 2.0 clients can't talk to anything older than a 1.4 server. If you ask a 2.0 client to fetch "STATUS", it will fail. You'll have to ask for "ups.status" instead.

Authors of separate monitoring programs should have used the 1.4 series to write support for the new variables and command names. Client software can easily support both versions as long as they like. If upsd returns *ERR UNKNOWN-COMMAND* to a GET request, you need to use REQ.

4.6 networkupstools.org

4.6.1 November 2003: a new URL

The bandwidth demands of a project like this have slowly been forcing me to offload certain parts to other servers. The download links have pointed offsite for many months, and other large things like certain UPS protocols have followed. As the traffic grows, it's clear that having the project attached to exploits.org is not going to work.

The solution was to register a new domain and set up mirrors. There are two initial web servers, with more on the way. The main project URL has changed from `http://www.exploits.org/nut/` to https://www.networkupstools.org. The actual content is hosted on various mirrors which are updated regularly with rsync, so the days of dribbling bits through my DSL should be over.

This is also when all of the web pages were redesigned to have a simpler look with fewer links on the left side. The old web pages used to have 30 or more links on the top page, and most of them vanished when you dropped down one level. The links are now constant on the entire site, and the old links now live in their own groups in separate directories.

4.7 Second major version

4.7.1 March 2004: NUT 2.0.0

NUT 2.0.0 arrived on March 23, 2004. The jump to version 2 shows the difference in the protocols and naming that happened during the 1.3 and 1.5 development series. 2.0 no longer ships with backwards compatibility code, so it's smaller and cleaner than 1.4.

4.8 The change of leadership

4.8.1 February 2005: NUT 2.0.1

The year 2004 was marked by a release slowdown, since Russell was busy with personal subjects. But the patches queue was still growing quickly.

At that time, the development process was still centralized. There was no revision control system (like the current Subversion repository), nor trackers to interact with NUT development. Russell was receiving all the patches and requests, and doing all the work on his own, including releases.

Russell was more and more thinking about giving the project leadership to Arnaud Quette, which finally happened with the 2.0.1 release in February 2005.

This marked a new era for NUT...

First, Arnaud aimed at opening up the development by creating a project on the Debian Alioth Forge. This allowed to build the team of hackers that Russell dreamed about. It also allows to ensure NUT's continuation, whatever happens to the leader. And that would most of all boost the projects contributions.