

Network UPS Tools Change Log

NUT project community contributors

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2.8.5.1685 2.8.5.1711- 1711+g1c4ab78c6	2026-10-30	Current release snapshot of Network UPS Tools (NUT).	
2.8.5	2026-04-07	Some changes to docs and recipes, especially for parallel builds; introduced a way to build with private shared libraries to save space. Updated NUT SEMVER definition some more, added a normalization and comparison script. Some rounds of code-hardening project. Added reference RPM/DEB recipes to test NUT iterations on OBS. Numerous driver updates, some new ones introduced.	JK
2.8.4	2025-08-07	Fixed a few regressions introduced by release v2.8.3. Some changes to docs and recipes, especially for parallel builds. Updated NUT SEMVER definition some more. Some rounds of code-hardening project. Numerous driver updates, some new ones introduced.	JK
2.8.3	2025-04-07	Some changes to docs and recipes, libupsclient API and functionality. Updated NUT SEMVER definition and added scripting around it. Groundwork for vendor-defined status and INSTCMD buzzwords like "ECO". Fixed some regressions and added improvements for certain new device series.	JK
2.8.2	2024-04-01	Some changes to docs and recipes, libnutscan API and functionality. Added nutconf (library and tool). Fixed some regressions and added improvements for certain new device series.	JK
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

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
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
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

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
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	Very detailed Change Log	1
2.1	2026-10-22 Jim Klimov <jimklimov+nut@gmail.com>	1
2.2	2026-10-21 sginestrini <sginestrini@users.noreply.github.com>	1
2.3	2026-10-21 nmbro <nmbro@users.noreply.github.com>	1
2.4	2026-10-21 Jim Klimov <jimklimov+nut@gmail.com>	2
2.5	2026-10-20 Luigi Barretta <thelugibarretta@gmail.com>	3
2.6	2026-10-20 Jim Klimov <jimklimov+nut@gmail.com>	3
2.7	2026-10-19 Jim Klimov <jimklimov+nut@gmail.com>	3
2.8	2026-10-18 nmbro <nmbro@users.noreply.github.com>	3
2.9	2026-10-18 Jim Klimov <jimklimov+nut@gmail.com>	3
2.10	2026-10-17 Jim Klimov <jimklimov+nut@gmail.com>	4
2.11	2026-10-16 user01010111 <lapses.50.booster@icloud.com>	4
2.12	2026-10-16 Jim Klimov <jimklimov+nut@gmail.com>	4
2.13	2026-10-15 user01010111 <lapses.50.booster@icloud.com>	5
2.14	2026-10-15 user01010111 <12504630+user01010111@users.noreply.github.com>	5
2.15	2026-10-15 nmbro <nmbro@users.noreply.github.com>	5
2.16	2026-10-15 Jim Klimov <jimklimov+nut@gmail.com>	6
2.17	2026-10-14 user01010111 <lapses.50.booster@icloud.com>	7
2.18	2026-10-14 nmbro <nmbro@users.noreply.github.com>	7
2.19	2026-10-14 Laela Zorana <zoranalaela9@gmail.com>	8
2.20	2026-10-14 Jim Klimov <jimklimov+nut@gmail.com>	8
2.21	2026-10-14 David Coulson <23066302+davidcoulson@users.noreply.github.com>	8
2.22	2026-10-13 nmbro <nmbro@users.noreply.github.com>	9
2.23	2026-10-11 user01010111 <lapses.50.booster@icloud.com>	12
2.24	2026-10-11 nmbro <nmbro@users.noreply.github.com>	13
2.25	2026-10-10 Jim Klimov <jimklimov+nut@gmail.com>	14
2.26	2026-10-09 Jakub Bartak <bartakjakub@gmail.com>	14
2.27	2026-10-08 nmbro <nmbro@users.noreply.github.com>	14
2.28	2026-10-07 nmbro <nmbro@users.noreply.github.com>	14
2.29	2026-10-07 Jim Klimov <jimklimov+nut@gmail.com>	15
2.30	2026-10-04 Warren Strong <warren.r.strong@gmail.com>	15
2.31	2026-10-03 d01 <d01.devel@gmail.com>	15
2.32	2026-10-03 Ronny Brandt <ronnybrandt@gmail.com>	16
2.33	2026-10-02 Michael Walker <34684351+walkermc20@users.noreply.github.com>	16
2.34	2026-10-02 Jim Klimov <jimklimov+nut@gmail.com>	16

2.35	2026-09-29 TomZanna <git@tomzanna.com>	17
2.36	2026-09-28 user01010111 <lapses.50.booster@icloud.com>	17
2.37	2026-09-27 nmbro <nmbro@users.noreply.github.com>	17
2.38	2026-09-27 Arcadiy Ivanov <arcadiy@ivanov.biz>	18
2.39	2026-09-26 d01 <d01.devel@gmail.com>	18
2.40	2026-09-25 Viktor Drobot <linux776@gmail.com>	19
2.41	2026-09-25 Jim Klimov <jimklimov+nut@gmail.com>	19
2.42	2026-09-24 user01010111 <lapses.50.booster@icloud.com>	19
2.43	2026-09-23 user01010111 <lapses.50.booster@icloud.com>	19
2.44	2026-09-23 Jordan Rodgers <com6056@gmail.com>	19
2.45	2026-09-23 Jim Klimov <jimklimov+nut@gmail.com>	19
2.46	2026-09-22 user01010111 <lapses.50.booster@icloud.com>	20
2.47	2026-09-21 Jim Klimov <jimklimov+nut@gmail.com>	20
2.48	2026-09-20 user01010111 <lapses.50.booster@icloud.com>	20
2.49	2026-09-20 user01010111 <12504630+user01010111@users.noreply.github.com>	23
2.50	2026-09-20 Jordan Rodgers <com6056@gmail.com>	23
2.51	2026-09-20 Jim Klimov <jimklimov+nut@gmail.com>	23
2.52	2026-09-19 user01010111 <lapses.50.booster@icloud.com>	24
2.53	2026-09-19 Jim Klimov <jimklimov+nut@gmail.com>	24
2.54	2026-09-18 antuspcm <18096106+antuspcm@users.noreply.github.com>	24
2.55	2026-09-18 Jim Klimov <jimklimov+nut@gmail.com>	24
2.56	2026-09-17 lfntn <6958805+lfntn@users.noreply.github.com>	24
2.57	2026-09-17 Jordan Rodgers <com6056@gmail.com>	25
2.58	2026-09-17 Jim Klimov <jimklimov+nut@gmail.com>	25
2.59	2026-09-16 Priyanshu <ankurdwivedi75@gmail.com>	25
2.60	2026-09-16 Jordan Rodgers <com6056@gmail.com>	25
2.61	2026-09-16 Jim Klimov <jimklimov+nut@gmail.com>	25
2.62	2026-09-15 user01010111 <lapses.50.booster@icloud.com>	26
2.63	2026-09-15 user01010111 <12504630+user01010111@users.noreply.github.com>	26
2.64	2026-09-15 nmbro <nmbro@users.noreply.github.com>	26
2.65	2026-09-15 Jordan Rodgers <com6056@gmail.com>	27
2.66	2026-09-15 Jim Klimov <jimklimov+nut@gmail.com>	27
2.67	2026-09-14 Jim Klimov <jimklimov+nut@gmail.com>	27
2.68	2026-09-13 user01010111 <12504630+user01010111@users.noreply.github.com>	28
2.69	2026-09-13 Jim Klimov <jimklimov+nut@gmail.com>	28
2.70	2026-09-12 Jim Klimov <jimklimov+nut@gmail.com>	28
2.71	2026-09-11 Jim Klimov <jimklimov+nut@gmail.com>	28
2.72	2026-09-10 user01010111 <12504630+user01010111@users.noreply.github.com>	28
2.73	2026-09-10 Jim Klimov <jimklimov+nut@gmail.com>	29

2.74	2026-09-10 Anton Popov <anton.a.popov@gmail.com>	29
2.75	2026-09-09 user01010111 <12504630+user01010111@users.noreply.github.com>	29
2.76	2026-09-09 Jim Klimov <jimklimov+nut@gmail.com>	30
2.77	2026-09-08 user01010111 <12504630+user01010111@users.noreply.github.com>	31
2.78	2026-09-08 Jim Klimov <jimklimov+nut@gmail.com>	31
2.79	2026-09-07 user01010111 <12504630+user01010111@users.noreply.github.com>	31
2.80	2026-09-07 Jim Klimov <jimklimov+nut@gmail.com>	31
2.81	2026-09-06 nmbro <nmbro@users.noreply.github.com>	32
2.82	2026-09-06 Laurent Bigonville <bigon@bigon.be>	33
2.83	2026-09-06 Jim Klimov <jimklimov+nut@gmail.com>	33
2.84	2026-09-05 user01010111 <lapses.50.booster@icloud.com>	33
2.85	2026-09-04 Jonathan Chavez <jonathangiancarlo1@gmail.com>	33
2.86	2026-09-04 Jiří Setnicka <setnicka@seznam.cz>	34
2.87	2026-09-02 Jim Klimov <jimklimov+nut@gmail.com>	34
2.88	2026-09-01 user01010111 <lapses.50.booster@icloud.com>	34
2.89	2026-09-01 Yifeng Li <tomli@tomli.me>	35
2.90	2026-09-01 Jim Klimov <jimklimov+nut@gmail.com>	35
2.91	2026-08-31 user01010111 <lapses.50.booster@icloud.com>	36
2.92	2026-08-31 Yifeng Li <tomli@tomli.me>	36
2.93	2026-08-31 Jim Klimov <jimklimov+nut@gmail.com>	37
2.94	2026-08-30 nmbro <nmbro@users.noreply.github.com>	38
2.95	2026-08-30 Lucas Bocchi <lucas.bocchi@gmail.com>	39
2.96	2026-08-30 Jim Klimov <jimklimov+nut@gmail.com>	40
2.97	2026-08-29 user01010111 <lapses.50.booster@icloud.com>	40
2.98	2026-08-29 nmbro <nmbro@users.noreply.github.com>	40
2.99	2026-08-28 nmbro <nmbro@users.noreply.github.com>	41
2.100	2026-08-28 Jim Klimov <jimklimov+nut@gmail.com>	42
2.101	2026-08-27 Jim Klimov <jimklimov+nut@gmail.com>	42
2.102	2026-08-27 Adam Bedard <adam.bedard@gmail.com>	42
2.103	2026-08-26 Jim Klimov <jimklimov+nut@gmail.com>	43
2.104	2026-08-26 Adam Bedard <adam.bedard@gmail.com>	43
2.105	2026-08-25 Jim Klimov <jimklimov+nut@gmail.com>	43
2.106	2026-08-25 Adam Bedard <adam.bedard@gmail.com>	43
2.107	2026-08-24 Jim Klimov <jimklimov+nut@gmail.com>	44
2.108	2026-08-23 Jim Klimov <jimklimov+nut@gmail.com>	44
2.109	2026-08-22 nmbro <nmbro@users.noreply.github.com>	44
2.110	2026-08-21 Jim Klimov <jimklimov+nut@gmail.com>	44
2.111	2026-08-20 nmbro <nmbro@users.noreply.github.com>	44
2.112	2026-08-20 Kyle Mason <masonkr@gmail.com>	45

2.113	2026-08-20	Jun Kurihara <junkurihara@users.noreply.github.com>	45
2.114	2026-08-20	Jim Klimov <jimklimov+nut@gmail.com>	46
2.115	2026-08-19	Jim Klimov <jimklimov+nut@gmail.com>	46
2.116	2026-08-18	Jim Klimov <jimklimov+nut@gmail.com>	47
2.117	2026-08-17	user01010111 <lapses.50.booster@icloud.com>	47
2.118	2026-08-17	Costa Tsousis <costa@netdata.cloud>	48
2.119	2026-08-16	Jim Klimov <jimklimov+nut@gmail.com>	48
2.120	2026-08-15	Jim Klimov <jimklimov+nut@gmail.com>	48
2.121	2026-08-14	user01010111 <lapses.50.booster@icloud.com>	48
2.122	2026-08-14	d01 <d01@users.noreply.github.com>	48
2.123	2026-08-14	Jim Klimov <jimklimov@gmail.com>	49
2.124	2026-08-14	Jim Klimov <jimklimov+nut@gmail.com>	49
2.125	2026-08-13	user01010111 <lapses.50.booster@icloud.com>	50
2.126	2026-08-13	Jun Kurihara <junkurihara@users.noreply.github.com>	50
2.127	2026-08-13	Jim Klimov <jimklimov+nut@gmail.com>	51
2.128	2026-08-13	Chris Allen <203867+chrisallen@users.noreply.github.com>	51
2.129	2026-08-12	Yuriy_Y <alliancesko@gmail.com>	51
2.130	2026-08-12	Falusi Tamás <tamas.falusi@gmail.com>	51
2.131	2026-08-11	Jim Klimov <jimklimov+nut@gmail.com>	52
2.132	2026-08-10	pixi-p <pixip@o2.pl>	52
2.133	2026-08-09	Jim Klimov <jimklimov+nut@gmail.com>	52
2.134	2026-08-06	Jim Klimov <jimklimov+nut@gmail.com>	52
2.135	2026-08-05	Jim Klimov <jimklimov+nut@gmail.com>	53
2.136	2026-08-04	Marcelo Pacheco <marcello.mpacheco@gmail.com>	53
2.137	2026-08-04	Jun Kurihara <junkurihara@users.noreply.github.com>	53
2.138	2026-08-04	Jim Klimov <jimklimov+nut@gmail.com>	53
2.139	2026-08-03	mkiera <itskiera20@gmail.com>	53
2.140	2026-08-03	Jun Kurihara <junkurihara@users.noreply.github.com>	53
2.141	2026-07-31	Jun Kurihara <junkurihara@users.noreply.github.com>	54
2.142	2026-07-30	Jun Kurihara <junkurihara@users.noreply.github.com>	54
2.143	2026-07-30	Jim Klimov <jimklimov+nut@gmail.com>	55
2.144	2026-07-28	John Grant <john.grant@dronedeploy.com>	55
2.145	2026-07-28	Jim Klimov <jimklimov+nut@gmail.com>	55
2.146	2026-07-27	Jim Klimov <jimklimov+nut@gmail.com>	56
2.147	2026-07-25	José San Juan <josesj@gmail.com>	56
2.148	2026-07-24	Lucas Willian Bocchi <lucas@lucas.inf.br>	56
2.149	2026-07-24	Jim Klimov <jimklimov+nut@gmail.com>	57
2.150	2026-07-23	Lucas Bocchi <lucas.bocchii@gmail.com>	57
2.151	2026-07-23	Jim Klimov <jimklimov+nut@gmail.com>	57

2.152	2026-07-22	dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>	57
2.153	2026-07-22	Jim Klimov <jimklimov+nut@gmail.com>	57
2.154	2026-07-21	Jim Klimov <jimklimov+nut@gmail.com>	58
2.155	2026-07-20	Jim Klimov <jimklimov+nut@gmail.com>	58
2.156	2026-07-19	Jim Klimov <jimklimov+nut@gmail.com>	59
2.157	2026-07-17	Jim Klimov <jimklimov+nut@gmail.com>	60
2.158	2026-07-15	Jim Klimov <jimklimov+nut@gmail.com>	60
2.159	2026-07-14	Jim Klimov <jimklimov+nut@gmail.com>	60
2.160	2026-07-13	Jim Klimov <jimklimov+nut@gmail.com>	61
2.161	2026-07-12	Jim Klimov <jimklimov+nut@gmail.com>	61
2.162	2026-07-09	Jim Klimov <jimklimov+nut@gmail.com>	61
2.163	2026-07-05	Programer <developer.sorli@gmail.com>	62
2.164	2026-07-03	Stephen Horvath <s.horvath@outlook.com.au>	62
2.165	2026-07-01	strasharo <3337997+strasharo@users.noreply.github.com>	62
2.166	2026-06-30	Jim Klimov <jimklimov@gmail.com>	62
2.167	2026-06-30	Jim Klimov <jimklimov+nut@gmail.com>	62
2.168	2026-06-29	Jim Klimov <jimklimov+nut@gmail.com>	63
2.169	2026-06-28	Jim Klimov <jimklimov+nut@gmail.com>	63
2.170	2026-06-26	juslex <66561713+juslex@users.noreply.github.com>	64
2.171	2026-06-26	Jim Klimov <jimklimov+nut@gmail.com>	64
2.172	2026-06-25	Thanh Nguyen <imscopilot@outlook.com>	65
2.173	2026-06-24	dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>	65
2.174	2026-06-24	Jim Klimov <jimklimov+nut@gmail.com>	65
2.175	2026-06-23	juslex <66561713+juslex@users.noreply.github.com>	66
2.176	2026-06-23	Jim Klimov <jimklimov+nut@gmail.com>	66
2.177	2026-06-23	Boris Shor <boris@bshor.com>	67
2.178	2026-06-22	Jim Klimov <jimklimov+nut@gmail.com>	67
2.179	2026-06-21	Jim Klimov <jimklimov+nut@gmail.com>	67
2.180	2026-06-21	Huang Rui <vowstar@gmail.com>	67
2.181	2026-06-21	Doug Nazar <nazard@nazar.ca>	67
2.182	2026-06-20	penghong123 <2835950872@qq.com>	68
2.183	2026-06-20	Doug Nazar <nazard@nazar.ca>	68
2.184	2026-06-19	Jim Klimov <jimklimov+nut@gmail.com>	68
2.185	2026-06-17	Jim Klimov <jimklimov+nut@gmail.com>	68
2.186	2026-06-16	Jim Klimov <jimklimov+nut@gmail.com>	68
2.187	2026-06-15	Jim Klimov <jimklimov+nut@gmail.com>	69
2.188	2026-06-14	mbv06 <mbv06.dev@gmail.com>	69
2.189	2026-06-12	Jim Klimov <jimklimov+nut@gmail.com>	69
2.190	2026-06-11	mbv06 <mbv06.dev@gmail.com>	69

2.1912026-06-11 Jim Klimov <jimklimov+nut@gmail.com>	69
2.1922026-06-10 dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>	69
2.1932026-06-09 mbv06 <mbv06.dev@gmail.com>	69
2.1942026-06-08 mbv06 <mbv06.dev@gmail.com>	69
2.1952026-06-08 Jim Klimov <jimklimov+nut@gmail.com>	70
2.1962026-06-07 Jim Klimov <jimklimov+nut@gmail.com>	70
2.1972026-06-06 Jim Klimov <jimklimov+nut@gmail.com>	70
2.1982026-06-05 Jim Klimov <jimklimov+nut@gmail.com>	70
2.1992026-06-03 Jim Klimov <jimklimov+nut@gmail.com>	70
2.2002026-06-02 Jim Klimov <jimklimov+nut@gmail.com>	71
2.2012026-06-01 Jim Klimov <jimklimov+nut@gmail.com>	71
2.2022026-05-31 Jim Klimov <jimklimov+nut@gmail.com>	71
2.2032026-05-31 Doug Nazar <nazard@nazar.ca>	71
2.2042026-05-28 Jim Klimov <jimklimov+nut@gmail.com>	71
2.2052026-05-27 Jim Klimov <jimklimov+nut@gmail.com>	71
2.2062026-05-26 Jim Klimov <jimklimov+nut@gmail.com>	72
2.2072026-05-25 Jim Klimov <jimklimov+nut@gmail.com>	72
2.2082026-05-24 Pedro Cunha <pedroagracio+nut@gmail.com>	72
2.2092026-05-22 Jim Klimov <jimklimov+nut@gmail.com>	73
2.2102026-05-21 Jim Klimov <jimklimov+nut@gmail.com>	73
2.2112026-05-20 Pedro Cunha <pedroagracio+nut@gmail.com>	73
2.2122026-05-20 Jim Klimov <jimklimov+nut@gmail.com>	73
2.2132026-05-19 Jim Klimov <jimklimov+nut@gmail.com>	73
2.2142026-05-18 Lukas Schmid <lukas.schmid@netcube.li>	74
2.2152026-05-18 Jim Klimov <jimklimov+nut@gmail.com>	74
2.2162026-05-17 juslex <66561713+juslex@users.noreply.github.com>	75
2.2172026-05-15 Lukas Schmid <lukas.schmid@netcube.li>	75
2.2182026-05-15 Jim Klimov <jimklimov+nut@gmail.com>	75
2.2192026-05-14 Lukas Schmid <lukas.schmid@netcube.li>	76
2.2202026-05-14 Jim Klimov <jimklimov+nut@gmail.com>	76
2.2212026-05-13 Lukas Schmid <lukas.schmid@netcube.li>	76
2.2222026-05-13 Jim Klimov <jimklimov+nut@gmail.com>	76
2.2232026-05-12 Lukas Schmid <lukas.schmid@netcube.li>	77
2.2242026-05-12 Jim Klimov <jimklimov+nut@gmail.com>	77
2.2252026-05-12 Doug Nazar <nazard@nazar.ca>	78
2.2262026-05-11 Lukas Schmid <lukas.schmid@netcube.li>	78
2.2272026-05-11 Jim Klimov <jimklimov+nut@gmail.com>	78
2.2282026-05-10 Jim Klimov <jimklimov+nut@gmail.com>	79
2.2292026-05-09 Jim Klimov <jimklimov+nut@gmail.com>	79

2.2302026-05-08 ostrich <570911+ostrich@users.noreply.github.com>	79
2.2312026-05-08 Jim Klimov <jimklimov+nut@gmail.com>	80
2.2322026-05-07 Jim Klimov <jimklimov+nut@gmail.com>	80
2.2332026-05-06 Jim Klimov <jimklimov+nut@gmail.com>	80
2.2342026-05-05 Jim Klimov <jimklimov+nut@gmail.com>	80
2.2352026-05-04 Jim Klimov <jimklimov+nut@gmail.com>	80
2.2362026-05-02 Jim Klimov <jimklimov+nut@gmail.com>	81
2.2372026-05-01 Jim Klimov <jimklimov+nut@gmail.com>	81
2.2382026-04-30 Jim Klimov <jimklimov+nut@gmail.com>	82
2.2392026-04-29 Jim Klimov <jimklimov+nut@gmail.com>	82
2.2402026-04-28 Jim Klimov <jimklimov+nut@gmail.com>	82
2.2412026-04-27 Jim Klimov <jimklimov@gmail.com>	82
2.2422026-04-27 Jim Klimov <jimklimov+nut@gmail.com>	83
2.2432026-04-26 Jim Klimov <jimklimov+nut@gmail.com>	83
2.2442026-04-25 Jim Klimov <jimklimov+nut@gmail.com>	83
2.2452026-04-24 Jim Klimov <jimklimov+nut@gmail.com>	84
2.2462026-04-23 Jim Klimov <jimklimov+nut@gmail.com>	85
2.2472026-04-22 Jim Klimov <jimklimov+nut@gmail.com>	85
2.2482026-04-21 Jim Klimov <jimklimov+nut@gmail.com>	86
2.2492026-04-20 Jim Klimov <jimklimov+nut@gmail.com>	86
2.2502026-04-19 Jim Klimov <jimklimov+nut@gmail.com>	87
2.2512026-04-18 Tom Eldon <tom@tomeldon.com>	87
2.2522026-04-18 Jim Klimov <jimklimov+nut@gmail.com>	87
2.2532026-04-16 Jim Klimov <jimklimov+nut@gmail.com>	87
2.2542026-04-15 Jim Klimov <jimklimov+nut@gmail.com>	88
2.2552026-04-14 Lukas Schmid <lukas.schmid@netcube.li>	88
2.2562026-04-14 Jim Klimov <jimklimov+nut@gmail.com>	88
2.2572026-04-14 Axel Gembe <axel@gembe.net>	89
2.2582026-04-13 Lukas Schmid <lukas.schmid@netcube.li>	89
2.2592026-04-13 Jim Klimov <jimklimov+nut@gmail.com>	90
2.2602026-04-12 Jim Klimov <jimklimov@gmail.com>	91
2.2612026-04-12 Jim Klimov <jimklimov+nut@gmail.com>	91
2.2622026-04-11 Lukas Schmid <lukas.schmid@netcube.li>	92
2.2632026-04-11 Jim Klimov <jimklimov+nut@gmail.com>	92
2.2642026-04-10 Jim Klimov <jimklimov+nut@gmail.com>	92
2.2652026-04-09 Jim Klimov <jimklimov+nut@gmail.com>	93
2.2662026-04-08 Jim Klimov <jimklimov+nut@gmail.com>	94
2.2672026-04-07 Jim Klimov <jimklimov+nut@gmail.com>	95
2.2682026-04-06 Jim Klimov <jimklimov+nut@gmail.com>	95

2.2692026-03-31 Jim Klimov <jimklimov+nut@gmail.com> 95

2.2702026-03-30 Jim Klimov <jimklimov+nut@gmail.com> 95

2.2712026-03-25 Jim Klimov <jimklimov+nut@gmail.com> 95

2.2722026-03-24 Jim Klimov <jimklimov+nut@gmail.com> 95

2.2732025-10-02 Jim Klimov <jimklimov+nut@gmail.com> 96

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.

2 Very detailed Change Log

This document intends to detail the change log for relatively recent work (roughly since the source code was tracked in Git).

Note

This change log section represents git commits in range `v2.8.5..HEAD` (commits `0e051f9f6..1c4ab78c6`).

2.1 2026-10-22 Jim Klimov <jimklimov+nut@gmail.com>

- `include/nut-pragmas-range-checks.h`: avoid `"-Wtautological-type-limit-compare"` with older-clang builds after all [#3770]
- `common/common.c`: `process_stat_symlink()`: range-check for `nbytes` is not needed now that it is natively `ssize_t` (so always does not exceed `SIZE_MAX/2`) [#3770]
- `tools/nut-scanner/nut-scanner.c`: `handle_arg_cidr()`: make use of `ifnamelen` to satisfy compiler warnings
- `common/common.c`: `xbasename_no_ext()`: fix destination size for `snprintf()` [GHSA-ghqr-q65h-w9wg]
- `include/nut-pragmas-range-checks.h`, `common/common.c`: fix pragma for range-check in `process_stat_symlink()` [#3770]
- `.github/scripts/appveyor/04-build_script.bat`, `.github/scripts/appveyor/07-test_script.bat`, `.github/scripts/appveyor/08-after_test.bat`, `.github/scripts/appveyor/*.bat`: report with a timestamp that "make something finished, pruning ccache" to explain the long delay without log output [#3794]
- `include/nut-pragmas-range-checks.h`, `common/common.c`: fix pragma for range-check in `process_stat_symlink()` [#3770]

2.2 2026-10-21 sginestrini <sginestrini@users.noreply.github.com>

- `docs/man/blazer-common.txt`, `drivers/blazer.c`: `drivers/blazer`: point the *no battery voltages* log at default.* names The blazer family reads the battery charge-range only via the `dstate`, i.e. from `default.battery.voltage.high/low` (or override.) **ups.conf entries; the plain names are reported values and storeval() rejects them as driver arguments. Neither the info-log hint nor the man page said so, which regularly leads users to put plain battery.voltage.** lines in `ups.conf` and hit a fatal *not a valid variable name* error. Extend the guesstimation INFO message to name the accepted keys, and say it explicitly in the `blazer-common` man page.

2.3 2026-10-21 nmbro <nmbro@users.noreply.github.com>

- `.github/scripts/appveyor/04-build_script.bat`, `.github/scripts/appveyor/07-test_script.bat`, `.github/scripts/appveyor/08-after_test.bat`, `NEWS.adoc`: `ci(appveyor)`: ccache cleanup after test and bundle A failed master build saves the build cache right after the failing step (`APPVEYOR_SAVE_CACHE_ON_ERROR`), but `"ccache -c"` only ran at the end of `04-build_script.bat` and in `on_success`. If the test or bundle step failed (e.g. by the CI deadline) after compiling more objects, the cache was saved without enforcing `max_size`, and could get near the per-item cache size limit again. Run `"ccache -c"` right after `"make check"` in `07-test_script.bat` and after `"make install-win-bundle"` in `08-after_test.bat`, whatever their result, and keep their exit codes (the 08 line drops `"set -e"` for that; the only other command there already handles its failure). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
-

- NEWS.adoc, appveyor.yml: ci(appveyor): no deadline for PRs, 57 min else PR builds do not save the build cache any more, so they need no early deadline to leave time for it: let them use the whole hour (the step scripts fall back to 3600 seconds when CI_DEADLINE_EPOCH is unset). For other builds, move the deadline from 51 to 57 minutes. Job logs show what still runs after it: 7z of the bundle 45s (at -mx=3), artifact upload and ccache stats \10s, cache save 25-40s; deadline-hit master jobs finished about 35s after it. That is \~1.5 minutes, so 57 leaves a margin of about 1.5 minutes before AppVeyor's 60-minute kill. Suggested in review of [PR #3790](#). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- tests/.gitignore, tests/Makefile.am, tests/nutsnprintftest.c: test: check C99 snprintf() truncation semantics New tests/nutsnprintftest checks that snprintf(), vsnprintf() and the NUT snprintf_dynamic(), vsnprintf_dynamic() and snprintfcat() helps NUL-terminate truncated output and return the would-be length. Fails on MinGW builds without the wincompat.h fix; should pass with any C99 libc. Refs: GHSA-ghqr-q65h-w9wg
- NEWS.adoc, include/wincompat.h: fix(wincompat): do not map snprintf() to _snprintf() Microsoft _snprintf()/_vsnprintf() return -1 and leave the buffer unterminated when the output does not fit; NUT code assumes C99 semantics. With MinGW-w64, snprintf() is a function, not a macro, so the #ifndef guards never stopped the mapping, and every file including wincompat.h got the Microsoft behavior. Keep the mapping only for MSVC before Visual Studio 2015 (_MSC_VER < 1900), which lacked C99 snprintf(). MinGW builds now use the C99 mingw_snprintf()/mingw_vsnprintf(). Refs: GHSA-ghqr-q65h-w9wg
- .github/scripts/appveyor/02-install.bat, .github/scripts/appveyor/03-before_build.bat, .github/scripts/appveyor/04-build_script.bat, .github/scripts/appveyor/05-after_build.bat, .github/scripts/appveyor/07-test_script.bat, .github/scripts/appveyor/08-after_test.bat, .github/scripts/appveyor/12-on_success.ps1, NEWS.adoc, appveyor.yml: ci(appveyor): fix cmd-eaten "\n", stop PR cache saves cmd.exe ignores single quotes, so "\n" inside bash -lc ... is cmd's operator: bash got only the left side, the rest ran in cmd or was silently dropped. - 05/07/08: use "if ! a ; then b ; fi" instead of "\n"; note the pitfall in appveyor.yml - 07: dump test logs in the same bash call, keep the real exit code (e.g. 124 from timeout), run make check with -j 2 - 08: drop dead symlink/find line; 7z -mx=3 -mmt=on (default level took ~3.8 min) - 04, new 12-on_success.ps1: run ccache -c before the cache save; ccache overshoots max_size (~120%), and an entry over AppVeyor's per-item limit is not saved, so the next build starts cold - 03: zero ccache stats so reports count this build only - 12-on_success.ps1: also post per-build ccache stats as a job message (shown in the GitHub notification of passed builds) - init: skip cache save in PR builds (one cache per project, each PR build evicted master objects and re-uploaded \~1 GB); save on error only for master pushes Assumes [#3786](#) lands with the suggested 02-install.bat line. Refs [#3786](#)

2.4 2026-10-21 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: refer to PR for appveyor update [[#3794](#)]
- tests/nutsnprintftest.c: ignore "-Wformat-truncation", buffer is big enough here
- tests/nutsnprintftest.c: quiesce format-nonliteral warning [[GHSA-ghqr-q65h-w9wg](#)]
- docs/man/blazer-common.txt, drivers/blazer.c: drivers/blazer.c: use MAN_SECTION_CMD_SYS for man page reference in the message [[#3792](#)]
- NEWS.adoc, docs/nut.dict: NEWS.adoc: update references to PRs of the past months
- NEWS.adoc: refer to ecoflow-hid PR [[#3788](#)]
- NEWS.adoc, include/wincompat.h: Cosmetic follow-up
- NEWS.adoc: rearrange numerous blocks about apcmicrolink driver into one [[#3719](#), [#3778](#)]
- include/nut-pragmas-type-limits-end.h, include/nut-pragmas-type-limits.h, include/nut-pragmas-unreachable-code.h: include/nut-pragmas*.h: update for older clang versions [[#3770](#)]
- .github/scripts/appveyor/02-install.bat: fix cmd\+bash syntax [[#3786](#), [#3790](#)] Co-authored-by: Nicolai Brogaard <nmb@minserver.dk>

2.5 2026-10-20 Luigi Barretta <thelugibarretta@gmail.com>

- NEWS.adoc, drivers/ecoflow-hid.c: drivers/ecoflow-hid.c: do not leak battery.runtime conversion result [#3787] ecoflow_battery_runtime returned a freshly xmalloc()ed HU_INFOSIZE buffer on every call. Its caller, ups_infoval_set() via hu_find_infoval(), treats conversion results as borrowed (built-in converters return static buffers, lookup tables return literals) and never frees them. As battery.runtime is HU_FLAG_QUICK_POLL, the driver leaked 128 bytes per poll: about 14 MB per day at pollinterval=1, 550 MB after 44 days on an EcoFlow River 3 Plus. Use a static buffer like the other conversion functions; the driver is single-threaded and dstate_setinfo() copies the value immediately. Analysis and patch prepared with the help of Claude Code (an AI coding assistant); verified on a production EcoFlow River 3 Plus (2.8.3-based build: VmData flat at 308 kB instead of \+48 kB per 300 s).

2.6 2026-10-20 Jim Klimov <jimklimov+nut@gmail.com>

- docs/man/Makefile.am: recover (once) from broken man page builds on freebsd with parallel builds
- clients/upsset.c: savesettings(): init desc=NULL to avoid bogus free() [#3668]
- tests/upsset-string-test.py: use subprocess.Popen() for finer debugging [#3668]
- tests/upsset-string-test.py: pass NUT_AUTHCONF_PATH to avoid confusion with testing user config [#3668]
- tests/upsset-string-test.py: honour NUT_DEBUG_LEVEL [#3668]
- tests/upsset-string-test.py: make sure upsset.cgi is always executed [#3668]

2.7 2026-10-19 Jim Klimov <jimklimov+nut@gmail.com>

- tests/upsset-string-test.py: mock web-server better [#3668]
- tests/upsset-string-test.py: use a common default_timeout=30 (sec) for stressed CI builders [#3668]
- tests/parseconf-test.c: fix variable declaration style [#3633]
- docs/man/Makefile.am: DOCBUILD_MAN: throttle parallel builds on FreeBSD :\
- docs/man/Makefile.am: refactor with DOCBUILD_MAN
- appveyor.yml, .github/scripts/appveyor/02-install.bat: use standard CCACHE_MAXSIZE envvar if possible [#3754]

2.8 2026-10-18 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, drivers/apcmicrolink-usb.c, drivers/apcmicrolink.c: apcmicrolink: USB: reap abandoned async transfers An async interrupt-IN transfer that does not confirm cancellation within the stop budget is marked abandoned and left for its callback to free. libusb_close() removes in-flight transfers from libusb's list without calling their callback, so after a close the callback never ran and the transfer and its buffer leaked. After the bus reset in microlink_usb_reset_and_reopen(), give the event loop a bounded chance to reap them before closing the handle. Start the pump thread before submitting the first transfer, so a pthread_create() failure leaves nothing outstanding and the abandon path is no longer needed at startup. Bump DRIVER_VERSION to 0.07. Closes: #3751 Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.9 2026-10-18 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: use "recognized" vs. "recognised" spelling, refer to PR [#3729]
 - NEWS.adoc: use "recognized" vs. "recognised" spelling [#3659]
-

2.10 2026-10-17 Jim Klimov <jimklimov+nut@gmail.com>

- docs/Makefile.am: refactor use of ASPELL_OUT_LOCATE_AWK [#3773] Avoid problems escaping double-quotes inside double-quoted backticked context.
- tests/NIT/nit.sh: do not create server.p12 twice [#3331]
- tests/NIT/nit.sh: check_NIT_certs_NSS(): consider more reasons to re-create the NSS DB from p12 copy [#3331]
- tests/NIT/nit.sh: log_info the failed certutil outputs [#3331]
- tests/NIT/nit.sh: trace unexpected WITH_SSL_SERVER/WITH_SSL_CLIENT values (e.g. none) [#3331]
- tests/NIT/nit.sh: Revert "tests/NIT/nit.sh: use NSS DBM (old, stable) format of crypto storage [#3331]" This reverts commit ce9ef8d39a04eb49a091bf03e3fc2c2f0f287aedi: Fedora completely dropped the DBM format support.
- tests/NIT/nit.sh: check_NIT_certs_NSS(): make sure we can read NSS DB files protected by passwords [#3331]
- tests/NIT/nit.sh: track creators of crypto format files on different systems [#3331] Help troubleshoot mismatches when many CI agents share the cache.
- tests/NIT/nit.sh: ensure .pwfile lacks trailing newline and is read same way on different systems [#3331]
- tests/NIT/nit.sh: use NSS DBM (old, stable) format of crypto storage [#3331]

2.11 2026-10-16 user01010111 <lapses.50.booster@icloud.com>

- clients/upsclient.c, common/common.c, include/common.h: clients, common: preserve warning and socket portability Guard the optional Clang warning query when __has_warning is unavailable. Restore select_connect's size_t interface and convert to socklen_t at connect(), so ordinary common.h consumers need no socket header. Developed with OpenAI Codex. Reviewed with Claude Opus 5.5 before the socket correction; the correction was subsequently validated on macOS and Linux.
- AGENTS.md: docs: clarify agent guidance for portability and review
- NEWS.adoc: NEWS: link Qx failed-claim correction to PR #3766 AI assistance: OpenAI Codex GPT-6 Astra.
- NEWS.adoc, docs/nutdrv_qx-subdrivers.txt, drivers/nutdrv_qx.c, drivers/nutdrv_qx.h, drivers/nutdrv_qx_innovart31.c, drivers/nutdrv_qx_innovart33.c, drivers/nutdrv_qx_santak.c: nutdrv_qx: keep failed mandatory claims from publishing state Validate cached mandatory values before publishing SANTAK, INNOVART31 and INNOVART33 claims. Retain query order and publish only after the complete claim, including the SANTAK topology check, succeeds. Addresses #3659. AI assistance: OpenAI Codex prepared the original implementation. OpenAI Codex GPT-6 Astra reviewed and validated the final proposal; Claude Code with Claude Opus 5.5 provided adversarial source review.
- docs/daisychain.txt, docs/man/upscli_get_or_create_ssl_context.txt: docs: preserve native OpenSSL references and clarify daisychain roots Retain SSL_CTX_new(3) references with a supplementary web link. Clarify internal device zero and root alarm/status names without changing the alarm counter or commit ordering.

2.12 2026-10-16 Jim Klimov <jimklimov+nut@gmail.com>

- tests/NIT/nit.sh: fix naming typo when generating upsmon.jks from NSS DB origin [#3331]
 - tests/NIT/nit.sh: refine creation of LOCKFILE
 - clients/upsclient.c, server/netssl.c: nss_password_callback(): report counterpart hostname in log [#3331]
 - tests/upsset-string-test.py: allow a more generous timeout for swamped CI builders
 - configure.ac: fix compiler location debug reporting when vars are not set
-

2.13 2026-10-15 user01010111 <lapses.50.booster@icloud.com>

- NEWS.adoc: NEWS: reference upslog startup leak fix PR
- NEWS.adoc: NEWS: reference [PR #3770](#)
- NEWS.adoc, clients/upslog.c: upslog: free parsed strings before reparsing monitored systems Release owned UPS name and hostname strings before the second startup parse replaces them. Keep the parse and its existing address handling. AI assistance: OpenAI Codex (GPT-6).
- NEWS.adoc, clients/authconf.c, clients/upsclient.c, common/common-nut_version.c, common/common.c, include/common.h: clients, common: fix strict Clang conversion warnings Use the API and index types in common helpers and make bounded client conversions explicit. Preserve the signed readlink result and check the symlink size before allocation. Keep the OpenSSL annotation exception local to its stack macros. Developed with OpenAI Codex (GPT-6).
- docs/configure.txt: keep C++ language names in body text Escape paired plus signs instead of using monospace markup for language names, following the developer documentation guidance. Prepared with OpenAI Codex: GPT-6 Astra for controller verification and GPT-6.1 Sol for implementation and local validation.
- NEWS.adoc: NEWS: reference [PR #3762](#) for SSL declaration fix
- NEWS.adoc, clients/upsclient.c: libupsclient: fix declaration order in SSL context setup

2.14 2026-10-15 user01010111 <12504630+user01010111@users.noreply.github.com>

- NEWS.adoc: NEWS: link the snmp-ups warning correction to [PR #3767](#) Add the assigned PR number to the existing IETF fallback warning entry. AI assistance: OpenAI Codex.
- NEWS.adoc, drivers/snmp-ups.c: snmp-ups: warn about IETF fallback only for the IETF mapping Compare the selected mapping name explicitly with ietf, so the fallback warning describes the mapping actually selected. Keep mapping selection and the startup model lookup unchanged. Bump the driver version and record the correction in NEWS. AI assistance: OpenAI Codex (gpt-6.1-sol and gpt-6-astra).

2.15 2026-10-15 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, m4/nut_check_aspell.m4: m4: find aspell resources in lib64 and X.Y dirs NUT_CHECK_ASPELL looked for the filter resources (to pass the location of "tex.amf" via --filter-path) only under "../lib" and "../share" from the aspell binary, in "aspell-<version>", "aspell-<version minus the last component>" or "aspell" subdirectories. On CentOS 7 the resources are in /usr/lib64/aspell-0.60 for aspell 0.60.6.1, so neither the "lib64" base nor the "aspell-0.60" subdirectory was tried, and the check fell back to the aspell built-in paths. Also try "../lib64" (after "../lib"), and the major.minor version subdirectory (before the plain "aspell" one), in loops rather than nested conditions. The order of the other candidates is not changed. Checked with configure: on Ubuntu with aspell 0.60.8.1 the detected paths are unchanged; with a fake CentOS 7 layout (aspell reporting 0.60.6.1, an empty "lib", resources in "lib64/aspell-0.60") the lib and tex paths are found where master found "none", and make spellcheck passes using that --filter-path. The other part of the issue, to enable the spellcheck only if aspell actually works, was done by the live check from commit e51932016: with a missing "en" dictionary (via ASPELL_CONF dict-dir), configure reports "no" for "if we have all the tools mandatory for documentation spell checks". Closes: [#2068](#) Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- Makefile.am, NEWS.adoc, ci_build.sh, docs/Makefile.am, docs/qa-guide.adoc: docs: locate spellcheck errors, annotate in GHA The spellcheck recipe reported unknown words as the aspell pipe-mode result lines, prefixed by "egrep -b" byte offsets into the aspell output, so finding the place in the source document took a search. Count the input lines in the aspell output instead (it ends the results for each line with an empty line), and report each unknown word as "file:line:col: <aspell result line>", with the file name relative to the root of NUT sources. The column offset aspell gives already matches the original line, as the recipe prefixes each line with a space. When running in GitHub Actions (GITHUB_ACTIONS=true), also print the errors as workflow commands, so they show as annotations on the source lines, with a few suggestions. The top-level spellcheck recipe collects them into a log file, reported once at the end, because the sequential retry after a failed parallel attempt reports the same errors again, and GitHub limits the count of annotations per step. Likewise, the summary re-run in ci_build.sh does not repeat them. No annotations are made with SPELLCHECK_ERROR_FATAL=no, as used by the distcheck steps after

the spelling was checked by another step. Also list the unknown words in the job summary (GITHUB_STEP_SUMMARY), with links to the source lines and a hint on updating docs/nut.dict. The ci_build.sh quiet first attempt (whose output is discarded) does not make annotations nor the summary either, so only its verbose attempt reports them. The awk program avoids ternary operators and hash characters, for older awk and make implementations. Checked with typos in NEWS.adoc, docs/man/upsc.txt and scripts/upsdrrvcctl/README.adoc: correct file, line and column, each annotation and summary table row printed once (also via BUILD_TYPE=default-spellcheck ci_build.sh), none with SPELLCHECK_ERROR_FATAL=no or without GITHUB_ACTIONS. A clean spellcheck passes, and distcheck-light passes with GITHUB_ACTIONS=true and a typo (as the dist worker runs it). In a test PR on a fork, typos showed as annotations on the changed lines, also in the old aspell-en 7.1 job. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

- .github/workflows/07-shellcheck.yml, .shellcheckrc, Makefile.am, NEWS.adoc, docs/nut.dict, docs/qa-guide.adoc, tools/Makefile.am, tools/shellcheck-ci.pl: ci: add ShellCheck check for changed scripts Add a "GHA-07: ShellCheck" workflow for pull requests, which runs ShellCheck on the shell scripts changed by the PR. The existing scripts have many known findings (about 2900 in the tree, 28 of them errors), so checking the whole tree would always fail. Instead, the new tools/shellcheck-ci.pl helper checks both the base and the PR versions of the changed scripts, and reports only new findings, matched by file, ShellCheck code and source line text (so moved code does not count as new). It fails for new "warning" or "error" level findings; new "note" level (info and style) ones are shown as notices. In GitHub Actions, the findings are shown as annotations on the changed lines and listed in the job summary, with links to the lines and the ShellCheck wiki. It can be run locally against any base revision, including uncommitted changes. The helper is in Perl, like tools/check-source-nonascii.pl (Perl is already needed for auto-gen.sh in Git workspaces). It only uses core Perl 5 modules, and parses the "gcc" output format of shellcheck, so no JSON module is needed on older systems. The workflow uses only actions/checkout (no new third-party actions) and compares the PR merge commit with its first parent. A new .shellcheckrc file disables the checks for deliberate portability idioms of NUT scripts: backticks (SC2006), x"\$var" comparisons (SC2268) and expr (SC2003). It is distributed like .editorconfig, to help the tarball users contribute too. Describe both in docs/qa-guide.adoc, replacing the note that the tool could be introduced at a later time. The .shellcheckrc backticks setting does for ShellCheck what the previous commit did for "make shellcheck" (issue #1058). Checked with the "build: keep full git version out of config.h" series (PR #3759): its first version is reported for one new SC2166 warning ([p -a q]) and the final one passes. Checked renamed, added, removed and non-shell files, a bad base revision, and the workflow with actionlint and zizmor. In a test PR on a fork, deliberate findings showed as annotations on the changed lines. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- Makefile.am, NEWS.adoc, docs/qa-guide.adoc: build: SHELL_PROGS for make shellcheck Let "make shellcheck" (check-scripts-syntax) check the POSIX shell scripts with each interpreter listed in the SHELL_PROGS envvar, like the NDE self-test already does, rather than only the system /bin/sh. Bourne-Again shell scripts are still checked with bash. The shell is named in the error message, so a failure is easy to tell apart. Also hide the "... obsolete, use \$(...)" warnings that e.g. ksh93 (the /bin/sh on illumos and Solaris) prints in syntax-check mode for each use of backticks, which NUT scripts use deliberately for portability: on current master, "ksh -n" printed 1442 of these for 47 scripts, which buried the 15 other warnings it had. The output is captured before filtering, so the syntax check exit code is kept. The filter uses \$(EGREP) with bracket expressions, rather than "grep -F", for older grep implementations. Checked with SHELL_PROGS="dash ksh bash" (ksh 93u\+m/1.0.8): all scripts pass, no backticks warnings are left, the other ksh warnings are still shown; and that syntax errors in a POSIX or a Bourne-Again shell script fail the check, naming the file and the shell. Spellcheck and distcheck-light pass. Closes: #1058 Closes: #1059 Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- NEWS.adoc, UPGRADING.adoc, configure.ac, docs/developers.txt, docs/nut-versioning.adoc, docs/nut.dict, tests/nut-driver-enumerator-test.sh, tools/gitlog2version.sh: build: add SEMVER_ACINIT for AC_INIT version Pass a new SEMVER_ACINIT query of gitlog2version.sh to AC_INIT instead of the bare SEMVER: it is the SEMVER for releases, and by the legacy convention a nominal "SEMVER.1" for any development iteration after that release, so the include/config.h macros still tell releases and development builds apart, but do not change between releases and keep build caches valid. It is built from the triplet, so a 4-component NUT_VERSION_DEFAULT (e.g. 2.8.5.1) does not get a second ".1" suffix. For pre-release tagged commits SEMVER is the upcoming release, so the VER50 triplet is used there (unless NUT_VERSION_FORCED_SEMVER is set). Checked with sh, dash and bash for git and NUT_VERSION_DEFAULT values (release, development, tarball and pre-release ones), and that config.h gets 2.8.5.1 while Makefiles keep the full 2.8.5.1503.2. Spellcheck passes with both current and old aspell-en 7.1. Update the gitlog2version test expectations in tests/nut-driver-enumerator-test.sh for the SEMVER_ACINIT value now included in the debug output, and add test cases for the new query (release, development, 4-component default and release candidate). Suggested in PR #3759 review. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.16 2026-10-15 Jim Klimov <jimklimov+nut@gmail.com>

- docs/daisychain.txt: restore a sentence lost in PR #3745

- NEWS.adoc: CI_AUTOGEN_STABLE_VERSION never appeared in a release so not worth mentioning in these notes [#3759]
- tools/gitlog2version.sh, tests/nut-driver-enumerator-test.sh: handle SEMVER_ACINIT corner case of X.Y.Z.0(.0...) [#3759]
- .gitignore: GitIgnore NIT.env only
- Jenkinsfile-dynamatrix: constrain dynamatrixGlobalState.labelDocumentationSpellchecker for NUT CI farm
- .gitignore: Revert "GitIgnore a (top_builddir)/NIT.env file that can be symlinked to simplify dev-testing" This reverts commit 28af610c189359ade33d96145e2aba6f097215ee: that file is now subject to make clean, and others succumb to git clean -fdX.
- docs/nut.dict: add www for some odd spellcheckers
- clients/upsclient.c, docs/man/microdowell.txt, drivers/nutdrv_qx_omron.c, tools/nut-scanner/nutscan-init.h: docs, code: settle on "initialization" not "initialisation" spelling
- docs/packager-guide.txt: update the table and preceding prose for "Package boundaries" section [#3748]
- docs/packager-guide.txt: restore commented "sandbox" discussions of historic context [#3748]
- docs/packager-guide.txt: fix commenting-away of "sandbox" discussions of historic context [#3748]
- docs/packager-guide.txt, docs/config-notes.txt: minor fixes for PR [#3748]
- NEWS.adoc: clarify the reason for a fix [#3762]

2.17 2026-10-14 user01010111 <lapses.50.booster@icloud.com>

- docs/man/belkinunv.txt, docs/man/microdowell.txt: docs: clarify Belkin wait mode and Microdowell shutdown delays Warn against setting wait in normal-monitoring configuration. Distinguish a -k request handled by a running driver from fresh shutdown-process initialisation when explaining configured and runtime delay values.

2.18 2026-10-14 nmbro <nmbro@users.noreply.github.com>

- INSTALL.nut.adoc, NEWS.adoc, appveyor.yml, ci_build.sh: ci: drop CI_AUTOGEN_STABLE_VERSION workaround With the full git version kept out of include/config.h, there is no need for ci_build.sh to force a shortened version into autogen.sh to keep ccache hits across pull requests. The workaround was AppVeyor only, still changed whenever master advanced, and also shortened the substituted @PACKAGE_VERSION@ (tarball, *.pc). Restore the plain autogen.sh call and drop the setting from appveyor.yml and the docs. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- INSTALL.nut.adoc, NEWS.adoc, UPGRADING.adoc, configure.ac, docs/developers.txt: build: keep full git version out of config.h include/config.h is included by every source file, and carried the full git-derived version (e.g. 2.8.5.1466.13) in PACKAGE_VERSION, PACKAGE_STRING and VERSION macros. It changed with every commit, so ccache direct mode missed for all objects on each new commit or fresh CI checkout. Without -g3, preprocessed mode mostly recovered the hits (no code uses these macros), but with -g3 (e.g. AppVeyor) the preprocessed output keeps all #define lines, so every object was recompiled. Pass the base SEMVER (e.g. 2.8.5) to AC_INIT, so config.h only changes with releases, and set the shell PACKAGE_VERSION/PACKAGE_STRING right after it back to the full VER50 value, so all @PACKAGE_VERSION@ substitutions (Makefiles, *.pc files, packaging recipes, tarball names, configure reports) keep the full version. Use AM_INIT_AUTOMAKE no-define (PACKAGE and VERSION macros are unused) and reset VERSION for distdir. Programs still report the full version via NUT_VERSION_MACRO in include/nut_version.h, consumed by two files. Checked with CCACHE_DEBUG=1 builds across two commits (WSL gcc 13, ubuntu-plucky gcc 14): config.h byte-identical, all objects direct hits except upsclient.o and common-nut_version.o. Also checked a dist tarball build (VERSION_DEFAULT), NUT_VERSION_FORCED, and autoconf 2.69 on CentOS 7 and Ubuntu Trusty. Docs go into the "Repetitive build/test optimizations" chapter of INSTALL.nut.adoc. Follow-up to PR #3754. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

- `.github/workflows/01-make-dist-worker.yml`: `ci(gha)`: old `aspell-en` without a Trusty container. Only the dictionary differs, so point the runner's own `aspell` at the `aspell-en 7.1` word lists via `ASPELL_CONF="dict-dir ..."`, instead of running the spellcheck in an `ubuntu:14.04` container. The upstream 7.1 tarball comes from the Azure Ubuntu archive mirror (plain HTTP, so it is pinned by sha256); its "make" just compiles the word lists with the installed `aspell`. Checked against CentOS 7 (`aspell 0.60.6.1`, `aspell-en 7.1` from EPEL): with an empty `docs/nut.dict`, both flag the same 11545 (file, word) pairs across all 282 checked files. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/CI.md`, `AGENTS.md`: `ci(gha)`: add `.github/CI.md` guidance, link from `AGENTS.md` Checks to run before pushing changes to `.github/` (`zizmor`, `actionlint`, `tsc --strict` for the helper scripts, and why there is no `package.json`), and the rules behind the current setup: untrusted input only as data, least privilege, avoid `pull_request_target`, trusted cache saves only, and no new third-party actions without a clear need (SHA-pinned). Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.19 2026-10-14 Laela Zorana <zoranalaela9@gmail.com>

- `NEWS.adoc`, `common/parseconf.c`, `tests/parseconf-test.c`: `common/parseconf.c`: warn when a hash glued to a word cuts a line [#3711] A config line such as `MONITOR ups@localhost 1 upsmon Pass#123 primary` is cut at the hash. Then `upsmon` reports a missing username, which does not point at the real cause. Print one warning to `stderr` with the line and word number. Leave out the word itself, so no part of a password reaches a log. Do it for file input only, so protocol traffic stays quiet. Parsing does not change, and the 23 existing `parseconf-test` cases still pass. Add checks to `tests/parseconf-test.c` for three inputs. They are a glued hash in a file, a spaced comment, and the glued line through `pconf_line`. Add a `NEWS.adoc` entry under common code. Closes: [#3711]

2.20 2026-10-14 Jim Klimov <jimklimov+nut@gmail.com>

- `docs/nut.dict`: clean away `genericups` (typo) [#3749]
- `clients/upsclient.c`: `upscli_get_or_create_ssl_context()`: determine `quiet_init_ssl` earlier now
- `clients/upsclient.c`: `upscli_get_or_create_ssl_context()`: fix reporting of calls in vain when not built with SSL to honour `NUT_QUIET_INIT_SSL` (in NIT tests)
- `NEWS.adoc`: update reference to `idowell-hid` fix [#3726]
- `NEWS.adoc`: move the note about second-level bullet points up, it drowned in the file
- `.github/FUNDING.yml`: Add funding option links via `Thanks.Dev` and `Ko-fi`
- `INSTALL.nut.adoc`, `docs/developers.txt`: introduce "Repetitive build/test optimizations" chapter [#3754]
- `NEWS.adoc`: document CI improvements [#3753, #3754, #3756]
- `Jenkinsfile-dynamatrix`: restore RISC-V tests on base C standard and `distcheck` [#3757]
- `NEWS.adoc`: document changes for `Jenkinsfile-dynamatrix` [#3757]
- `Jenkinsfile-dynamatrix`: introduce `dynacfgPipeline.excludeCombos_RISCV_DEFAULT` et al [#3757]
- `Jenkinsfile-dynamatrix`: modernize `dynacfgPipeline[spellcheck]` definition to match evolution in the `jenkins-dynamatrix` library

2.21 2026-10-14 David Coulson <23066302+davidcoulson@users.noreply.github.com>

- `NEWS.adoc`, `drivers/apc-ats-mib.c`: `drivers/apc-ats-mib.c`: map output phase table for current, power and load APC ATS models without the output bank table, such as the AP4452A, report output data only in `atsOutputPhaseTable`. Map `atsOutputCurrent` to `output.current`, `atsOutputPower (VA)` to `ups.power` and `atsOutputPercentPower` to `ups.load`, and read `atsOutputVoltage` for `output.voltage` before the existing `atsOutputBankOutputVoltage` entry, so devices with the bank table keep publishing that value. Columns reading -1 (unsupported) are skipped. Bump MIB version to 0.61. Tested against an AP4452A (AOS 3.2.0.7) over repeated update polls.

2.22 2026-10-13 nmbro <nmbro@users.noreply.github.com>

- `.github/workflows/01-make-dist-worker.yml`, `.github/workflows/05-codeql.yml`: `ci(gha)`: use `apt-get`, not `apt`, in workflows The `apt` front-end is meant for interactive use: it warns that its command line interface is not stable for scripts, and its output and behavior may change between releases. Use `apt-get` in the prerequisite steps of the dist worker and CodeQL workflows, with an explicit `-y` (instead of relying on the runner image's `APT::Get::Assume-Yes`). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/workflows/01-make-dist-worker.yml`: `ci(gha)`: pin dist worker bases, faster apt Run the dist worker jobs on `ubuntu-24.04` instead of `ubuntu-latest` (which moves to Ubuntu 26 soon), and pin the `ubuntu:14.04` image of the old-dictionary spellcheck job by digest, so the toolchain and the `aspell-en` dictionaries only change when bumped deliberately. The Trusty archive is frozen, so this also makes that `apt-get` step deterministic. Install the host prerequisites via `eatmydata (dpkg fsync())` is not needed on a throw-away runner) and without recommended packages, to speed up that step. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/workflows/01-make-dist-worker.yml`, `Jenkinsfile-dynamatrix`, `NEWS.adoc`, `ci_build.sh`: `ci`: spellcheck only in dedicated CI steps Pass `--disable-spellcheck` to the Jenkins autotools build matrices and to `ci_build.sh BUILD_TYPE=default-withdoc*` (the only default-* build types that did not disable it yet), so their "make check" no longer runs the docs spellcheck. A failing matrix build is then clearly not about spelling, and the jobs save a little time. Local builds keep the "configure --enable-spellcheck=auto" default. The GitHub Actions dist tarball worker now runs the spellcheck as explicit steps: with the current `aspell-en` of the runner, and via "docker run" in Ubuntu 14.04 for `aspell-en 7.1`, the same old dictionary as CentOS 7 (EPEL), which rejects some spellings that newer dictionaries accept. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `docs/nut.dict`: docs: add `withdoc` to `nut.dict` The `NEWS.adoc` entry for the spellcheck CI change names the `BUILD_TYPE=default-withdoc*` scenarios, which `aspell` flags. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `AGENTS.md`, `docs/man/nutdrv_qx.txt`: docs: artefact → artifact The `aspell-en 7.1` dictionary (CentOS 7, Ubuntu Trusty) rejects the British spelling, so it would fail the new old-dictionary spellcheck job. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/dependabot.yml`, `.github/workflows/01-make-dist-report.yml`, `.github/workflows/01-make-dist-worker.yml`, `.github/workflows/01-make-dist.yml`: `ci(gha)`: correct GHA-01 and Dependabot comments - Artifact name step no longer reads the PR branch name. - Old artifacts are not deleted for master or tags. - A manual worker run on master with an untrusted ref would use the master cache scope. - Non-GitHub actions are SHA-pinned now, only GitHub's own use tags. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/scripts/dist-tarballs-report.js`, `.github/scripts/pr-comments.js`, `.github/workflows/01-make-dist-report.yml`, `.github/workflows/01-make-dist-worker.yml`, `.github/workflows/01-make-dist.yml`, `.github/zizmor.yml`: `ci(gha)`: GHA-01 builds on `pull_request`, reports via `workflow_run` Under `pull_request_target`, the PR build job ran in the base branch cache scope: not saving the `ccache` there was not enough, as PR code can get the job's Actions runtime token and write cache entries for later master builds (e.g. a `ccache.conf` with a `prefix_command`). Split GHA-01 into a build workflow on `pull_request` (read-only token, the PR's own cache scope, also for forks), and a "GHA-01: Tarballs Report" workflow on `workflow_run` (default branch code only, with the write permissions) for the PR comment, commit status and artifact clean-up. It uses the build run only as data: the PR is looked up by head owner/branch and must match the run's head commit (needed anyway, as `workflow_run` lists no PRs from forks), and the artifact name is computed there. The worker puts the link in its job summary. This also drops `pull_request_target`, which GitHub blocks by default on public repositories from November 2, 2026. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/scripts/appveyor/02-install.bat`, `.github/scripts/appveyor/03-before_build.bat`, `.github/scripts/appveyor/04-build_script.bat`, `.github/scripts/appveyor/05-after_build.bat`, `.github/scripts/appveyor/07-test_script.bat`, `.github/scripts/appveyor/08-after_test.bat`, `.github/scripts/appveyor/08-after_test.ps1`, `appveyor.yml`, `docs/qa-guide.adoc`: `appveyor.yml`: move step scripts to `.github/scripts/appveyor/` Keep each AppVeyor step section in a script named after it, under `.github/scripts/appveyor/`: `.bat` for `cmd` steps, `.ps1` for `ps`. Script names are numbered by the position of their section among the AppVeyor pipeline script phases (02-install, 03-before_build, ..., 08-after_test), so the numbers stay stable if more phases get used later. The commands are moved verbatim (same order). AppVeyor checks the exit code of each line in inline scripts, but only of the last one for a called `.bat`, so each command is followed by an errorlevel check to keep failing fast (with the original exit code). `init` stays inline, as it runs before the repository is cloned. The trailing `ccache` stats of `after_test` are now reported by the `.bat` before (not after) the artifact link message. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

- `scripts/Windows/dllldd.sh`: filter more system DLLs, `\%s.dll` Add `combase`, `hid`, `setupapi`, `shfolder`, `wship6` and `xmllite` to the list of OS-provided DLLs, which were reported as not found (strings matcher) for every NUT for Windows bundle on AppVeyor. The `%s.dll` format-string filter missed a `\%s.dll` hit (backslash path separator in a format string); read later stripped the backslash, so it was reported as not found as `%s.dll`. Allow a leading path. Checked against the AppVeyor build 5659 bundle: identical DLL list, these 7 warnings gone, no new ones. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `scripts/Windows/dllldd.sh`: scan only data sections `strings` scanned whole files, including debug sections: for the NUT for Windows bundle (616MB of -g3 binaries) that produced 561MB of text, all piped through `tr/grep/sort`, which dominated the DLL scan of `make install-win-bundle` (~2 min in the first pass on AppVeyor). With `strings -d` (data sections only, as older `binutils` did by default) it is 2.8MB, and the same candidate DLL names. Fall back to a whole-file scan if `strings` lacks the option. Similarly use `objdump -p` (private headers, with the PE import table and ELF `NEEDED` entries) instead of `-x`, which also dumped symbol tables and relocations (37MB vs 9.8MB of text here). Checked against the AppVeyor build 5659 bundle: identical results and warnings to the original script. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `tests/NIT/nit.sh`: fix test certificate cache never repopulating When cached certificates failed `check_NIT_certs`, the cache dir was removed and immediately re-created empty. The cache is populated only if that dir is absent, so it stayed empty forever, and every run (seen on AppVeyor) generated new certificates after "GRAVE PROBLEM" noise. Do not re-create the cache dir there. Also fix a `|| \` typo (vs. `|| \` line continuation), which ran a command named `\` and then the `cp -L` fallback unconditionally. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `ci_build.sh`: correct `CI_AUTOGEN_STABLE_VERSION` comment The cache misses come from the `ccache` direct mode hashing `config.h` contents, not from -g3 (`ccache` does not treat -g3 specially). Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `appveyor.yml`: fail before the job timeout, to save cache AppVeyor kills the job at 60 minutes without saving the build cache, so a cold-`ccache` build never warms up (and an interrupted cache save in build 5650 lost the `ccache` entry entirely). Set a deadline at 51 minutes after job start in `init`, and run the long bash steps (`prereqs`, `build`, `check`, `spellcheck`, `bundle`) under `timeout` with the time left. If they would run past it, the job fails instead, and `APPVEYOR_SAVE_CACHE_ON_ERROR` keeps what was built, with at least ~5 minutes left for 7z, artifacts and the cache save. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `ci_build.sh`, `appveyor.yml`: stable `AC_INIT` version for `ccache` `AC_INIT` takes the version from `gitlog2version.sh`, which includes the number of commits in a PR (e.g. 2.8.5.1466.13), and puts it into `config.h` as `PACKAGE_VERSION`. With -g3 every object file embeds all macros it saw, so each new commit count meant a full rebuild on a 2-CPU AppVeyor worker (~50 min), and `ccache` hits only for PRs with a commit count seen before. Add opt-in `CI_AUTOGEN_STABLE_VERSION=true` to drop that component via `NUT_VERSION_FORCED` for `autogen.sh` only, and enable it for AppVeyor. `make` still generates `nut_version.h` from Git, so programs report the actual version. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `appveyor.yml`: store cache entries uncompressed, cap `ccache` size Single-threaded LZMA at `-mx=6` took 1-2 minutes per run to re-pack the already-compressed `ccache` entry. Store entries as-is instead, and cap `ccache` at 750MB to stay below the 1GB build cache limit for PRs along with the other cached entries. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `scripts/Windows/dllldd.sh`: list search dirs once per call `dllldd_with_tools()` ran a `find` per DLL name over `DESTDIR` and over each input directory. Once recursion reaches e.g. `/mingw64/bin`, this means hundreds of walks over a large directory, which is slow on MSYS2 (about 5 minutes of `make install-win-bundle` in AppVeyor CI). List those locations once per call and match base names with `awk`. Listing order is kept, so the first hit is the same file as before. Use of coding helper tools and AI disclosed: Claude Code was used. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- `.github/workflows/08-PyNUTClient.yml`, `.github/zizmor.yml`: `ci(gha)`: pin non-GitHub actions to commit SHAs Actions from GitHub's own organizations (`actions/`, `github/`) may stay referenced by tags (their major version tags move to new releases by design, so this trusts those organizations, which run the workflows anyway). Pin all other actions to a full commit SHA, with the version in a comment (which Dependabot updates too), as their tags or branches can be moved to other code: - `pypa/gh-action-pypi-publish @release/v1` (a branch) → `v1.14.2` Also use `actions/checkout@v7` instead of `@master` (a moving branch) in the `PyNUTClient` workflow, like elsewhere. The `zizmor` check now enforces this policy (ref-pin for `actions/*` and `github/*`, hash-pin for all others) via `.github/zizmor.yml`. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

- `.github/workflows/06-zizmor.yml`, `.github/zizmor.yml`: `ci(gha)`: add zizmor check for `.github` changes Add a "GHA-06: zizmor" workflow, which runs the zizmor static analysis of GitHub Actions workflows (a pinned version from PyPI, no third-party action) for pushes to master and PRs changing `.github/` files. It shows findings as annotations, and fails for new ones of at least "medium" severity. It runs on `pull_request` with a read-only token. The deliberate choices of this repository are recorded, with reasons, in `.github/zizmor.yml`: actions are referenced by tags (updated by Dependabot), and GHA-01 uses `pull_request_target` on purpose. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/dependabot.yml`: `ci(dependabot)`: 7 day cooldown for action updates Only propose GitHub Actions versions published at least a week ago, so that a compromised release is more likely to be found and pulled before it is picked up, since actions are referenced by tags (zizmor "dependabot-cooldown" finding). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/08-PyNUTClient.yml`: `ci(gha)`: no git credentials kept in PyNUTClient The PyNUTClient publishing workflow does not use git credentials after its checkout, so do not persist them in the working copy (zizmor "artipacked" finding). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/01-make-dist.yml`: `ci(gha)`: GHA-01 comments match what it does Update comments of the GHA-01 workflow which no longer matched it: - `pull_request_target` runs the workflow (and its `.github/scripts/`) from the default branch, not the PR target branch (GitHub changed that in late 2025), and the permissions are now per job; - the old artifacts clean-up code is in `deleteOldArtifacts()` now; - the workflow also runs weekly and manually; - note again what the "Preparing ..." PR comment step is about. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/scripts/dist-tarballs-report.js`, `.github/scripts/pr-comments.js`, `.github/workflows/01-make-dist.yml`: `ci(gha)`: GHA-01 scripts in `.github/scripts/` Move the inline actions/github-script code of GHA-01 into two plain JavaScript modules, which the steps load with `require()`: - `pr-comments.js`: generic helpers to create or update one comment of a workflow on the PR (found by a hidden marker), and to delete PR comments of another account (used for old AppVeyor CI comments); - `dist-tarballs-report.js`: tarball artifact lookup and clean-up, and reporting the link as PR comment, job summary and commit status. The PR comment logic is no longer repeated, and the code can be read, linted and type-checked as normal files: they have `//@ts-check` with JSDoc types, and pass `tsc --strict` (see the comment at the top of `pr-comments.js` for how to check them). No build step is needed. The jobs with write permissions get the modules with a sparse checkout of `.github/scripts` from their default ref, i.e. the trusted workflow commit (also for `pull_request_target` events), never the PR source. Old artifacts are now deleted with awaited API calls, listing all pages of artifacts with that name, instead of the first page only. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/01-make-dist-worker.yml`: `ci(gha)`: save dist worker ccache only for master The worker restored and saved its ccache with `actions/cache` on every run, including builds of PR code. Under `pull_request_target` those run in the base (master) branch cache scope, so a PR could plant a poisoned cache that later master branch builds restore and use. Split it into `actions/cache/restore` (always) and `actions/cache/save`, which only runs for pushes to master and the scheduled runs (reusing the key computed when restoring, as its `.timestamp-init` file is gone by then). PR builds still benefit from the master branch cache. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/00-preclean.yml`, `.github/workflows/01-make-dist.yml`: `ci(gha)`: move AppVeyor comment cleanup to GHA-01 Delete outdated AppVeyorBot comments on PRs in the "prepare" job of GHA-01 with a few lines of `actions/github-script`, and remove the "GHA-00: Pre-clean" workflow, which did only that, using a third-party action (maheshrayas/action-pr-comment-delete) with the repository hard-coded, and broader permissions than needed (contents: write, checks: write) in a `pull_request_target` workflow. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/01-make-dist-worker.yml`, `.github/workflows/01-make-dist.yml`: `ci(gha)`: GHA-01 calls worker as reusable workflow Instead of dispatching the "Worker" workflow as a separate run with `benc-uk/workflow-dispatch`, and keeping a job polling for its status all the time it runs (doubling the runner time used), call it as a reusable workflow (it already declared `workflow_call`), so it runs as part of this workflow run. GHA-01 now has three jobs, each with only the permissions it needs: prepare (artifact name, PR comment; pull-requests: write), build (the worker touching untrusted code; contents: read) and report (artifact lookup and cleanup, PR comment, summary, commit status; actions/pull-requests/statuses: write). The report job also runs when the worker failed, and the artifact is looked up in this run only. The workflow no longer needs "actions: write" to start runs. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
 - `.github/workflows/01-make-dist.yml`: `ci(gha)`: GHA-01 artifact name without action Compute the artifact name in a shell step instead of with `bluwy/substitute-string-action`: PR-<number>, else the branch or tag name, with characters that are not valid in artifact names replaced by `"_"` (a superset of the `" "` and `"/"` handled before, so usual names do not change). The values come in via env, since a PR branch name is author-controlled; it was expanded into a shell command line before. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
-

- `.github/workflows/01-make-dist.yml`: `ci(gha)`: GHA-01 PR comment, summary, status Replace thollander/actions-comment-pull-request with a small actions/github-script that finds the PR comment by a hidden marker (also taking over comments posted by that action before) and updates it, or creates one. The final comment also says when the worker run failed or produced no tarballs, instead of leaving "Preparing ..." in place. Values are passed via env, not expanded into script code. Replace the curl-made check run, whose `details_url` is ignored for `GITHUB_TOKEN` and whose name got an unpredictable prefix, with a commit status "GHA-01: Tarballs / download link": its `target_url` is honored, so "Details" leads to the artifact right away. This needs "statuses: write" instead of "checks: write". The link is also written to the job summary of the run. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.23 2026-10-11 user01010111 <lapses.50.booster@icloud.com>

- NEWS.adoc: docs: reference [PR #3748](#) in release notes
- NEWS.adoc: docs: reference [PR #3749](#) in release notes
- INSTALL.nut.adoc, NEWS.adoc, docs/config-notes.txt, docs/man/index.txt, docs/nut.dict, docs/packager-guide.txt, docs/qa-guide.adoc, docs/support.txt: docs: complete operator and contributor guidance The packager guide contains an unfinished package split and configure example, and the QA guide has empty self-test and Jenkins sections. This replaces those sections with deployment roles, build-content guidance and source-based introductions to the test and pipeline configuration. It also clarifies service-managed and manual startup, explains multi-token UPS status, adds a reminder to redact support data, directs Windows readers to current archives while retaining historical MSI instructions, and indexes existing authentication, SSL and library logging manuals. Three Jenkins identifiers are added to the spelling dictionary. Program behaviour is unchanged. Validation: single and chunked guides, the Windows installation text, the support page, manual index links and their real manual pages, scoped spellcheck, ASCII and whitespace checks, patch application, and make -j2 distcheck-light. AI assistance: GPT-6 Astra coordinated and reviewed the final change; GPT-6.1 Sol performed source checks, drafting and local validation.
- NEWS.adoc, docs/cables/ge-imv-victron.txt, docs/man/adelsystem_cbi.txt, docs/man/generic_modbus.txt: docs: describe Modbus timeouts and fix a cable driver name Document the existing response and byte timeout settings for generic_modbus and adelsystem_cbi, including their seconds and microseconds components, defaults and the upper limit on each microseconds component. This makes the settings available to users without implying four separate timeouts. Also correct genericsups to genericups in the GE/IMV/Victron cable reference. Source checks confirm the option names, defaults and component mapping for both supported libmodbus API forms in the drivers. Real HTML and man-page builds, rendered-text checks and spellcheck pass. The corrected cable reference also renders through the include layout in [PR #3742](#). The distcheck-light packaging check passes. AI assistance: OpenAI Codex with GPT-6 Astra for coordination and final review, and GPT-6.1 Sol for drafting and validation.
- NEWS.adoc: docs: reference [PR #3745](#) in release notes
- NEWS.adoc: docs: reference [PR #3743](#) in release notes
- NEWS.adoc: docs: reference [PR #3744](#) in release notes
- NEWS.adoc, docs/daisychain.txt, docs/design.txt, docs/developers.txt, docs/hid-subdrivers.txt, docs/maintainer-guide.txt, docs/man/libnutclient_general.txt, docs/man/upscli_get_default_connect_timeout.txt, docs/man/upscli_get_or_create_ssl_context.txt, docs/man/upscli_set_ssl_context.txt, docs/new-clients.txt, docs/nutdrv_qx-subdrivers.txt, docs/snmp-subdrivers.txt, docs/snmp.txt: docs: align developer guidance and client API references with source Correct client API declarations that conflict with the public headers: `upscli_get_default_connect_timeout()` returns `void` and writes through its argument, while `strarr_alloc()` takes `size_t` and initialises the allocated pointers to `NULL`. This makes the documented declarations usable against the current headers. Correct the daisychain alarm guidance: the count aggregates alarms within an update cycle, so an empty later device does not cancel an earlier device's alarm. Include status-buffer initialisation and the `status_commit()` then `alarm_commit()` order used by the SNMP driver to retain the aggregate alarm state. Align Qx and SNMP mapping references with their actual types. Repair protocol, API, appendix and upgrade-file references, point SSL context references to the official OpenSSL documentation, and direct SNMP extension readers to the existing guide. Remove GitHub SVN instructions following the service's retirement; correct the USB expansion and frame the listed HID vendors as examples. Validation passed: original API declarations fail against the current headers and corrected declarations compile; extracted daisychain helpers confirm alarm-count behaviour. Source checks, real renders of all changed manuals, single-page and chunked guides, rendered links, spelling, ASCII/whitespace checks and make -j2 distcheck-light pass. The official OpenSSL destination returned HTTP 200 with the correct documentation title. The unchanged `libnutclient_tcp`

warning and 27 baseline maintainer-guide spelling diagnostics reproduce unchanged. AI assistance: GPT-6 Astra coordinated the work and performed the final review; GPT-6.1 Sol assisted with checks and drafting.

- NEWS.adoc, docs/FAQ.txt, docs/config-notes.txt, docs/man/nut-scanner.txt, docs/man/nutconf.txt, docs/man/upslog.txt, docs/-man/upsmon.conf.txt, docs/man/upsmon.txt, docs/outlets.txt, docs/security.txt: docs: correct operator guidance and complete notification references Some operator guidance differs from the current implementation. The FAQ recommends increasing DEADTIME to suppress polling warnings, but this delays the protection applied when a UPS last known to be on battery becomes unreachable. Point readers to POLLFAIL_LOG_THROTTLE_MAX and NOCOMMWARNTIME for the corresponding log and notification controls. Clarify that losing UPS communication does not immediately shut down every computer: the last known UPS state, DEADTIME and configured power requirements determine the response. Keep the warning about powering communication equipment from managed outlets. Also correct the certificate-expiry comparison, remove the startup-only claim about upsd device definitions, and align the debug stream and polling-interval descriptions with the implementation. Document the existing scanner thread limit, complete the notification lists and replace the outdated SSL-directive disclaimer with links to the configuration manuals. This changes documentation only. Validation passed: source and notification-inventory checks, real HTML and roff renders of all changed manuals, single-page and chunked guides, affected links, spelling, ASCII-whitespace checks and make distcheck-light. The unchanged libnutclient_tcp rendering warning reproduces on the baseline. AI assistance: GPT-6 Astra coordinated the work and performed the final review; GPT-6.1 Sol assisted with checks and drafting.
- NEWS.adoc, docs/man/belkinunv.txt, docs/man/microdowell.txt, docs/man/nhs_ser.txt, docs/man/usbhid-ups.txt, docs/nut.dict: docs: correct and complete four driver option references The driver manuals omit existing settings or describe names that differ from the registered options. Correct usbhid-ups's onlinedischarge_onbattery spelling and replace belkinunv's claim that it has no configuration settings with a reference to its documented options. Document Microdowell shutdown and restart delays, including their defaults and bounds. Distinguish the stored shutdown delay from the minimum six-second delay used when sending a command. Explain that runtime updates affect the running instance and configured values are loaded again on restart, including a separate -k invocation; this matters when preparing a shutdown command. Document NHS model-name overrides and bypass-alarm control, including their limits and defaults. Add the two existing option names to the spelling dictionary. No driver behaviour changes. Validation passed: registered-option and implementation checks, real HTML and roff renders of all four changed manuals, the Belkin section link, spelling, ASCII/whitespace checks and make distcheck-light. Source-extracted checks exercise Microdowell defaults, bounds and command construction, and NHS model and alarm handling with surrounding operations stubbed. No physical UPS was exercised. The unchanged libnutclient_tcp rendering warning reproduces on the baseline. AI assistance: GPT-6 Astra coordinated the work and performed the final review; GPT-6.1 Sol assisted with checks and drafting.
- NEWS.adoc: reference [PR #3742](#)
- NEWS.adoc: reference [PR #3741](#)
- NEWS.adoc: reference [PR #3740](#)
- NEWS.adoc, docs/Makefile.am, docs/cables.txt: docs: include text cable references in generated guides Include the eight existing cable reference files as literal text in the cable guide and user manual. Keep ASCII pin diagrams and contributor attribution intact, and place PDF page breaks at diagram boundaries. Track the included references as direct dependencies of both documents. Fixes [#1367](#).
- NEWS.adoc, drivers/main.c: drivers: show identical configuration options only once in help Skip earlier entries with the same name, complete variable type and help description when printing driver usage. Keep registration and configuration handling intact, along with the parsable -L output. This removes repeated nutdrv_qx protocol options while retaining all distinct descriptions and types. Fixes [#2064](#) Assisted by OpenAI Codex: GPT-6 Astra for coordination and verification, and GPT-6.1 Sol for implementation and local validation.
- NEWS.adoc, docs/configure.txt, docs/nut.dict: docs/configure.txt: document debug and CppUnit options Document --with-debuginfo and --enable-cppunit defaults, supported settings and dependency checks for developers and troubleshooters. Refs [#2311](#). Prepared with OpenAI Codex: GPT-6 Astra for controller oversight and verification, and GPT-6.1 Sol for implementation and local validation.

2.24 2026-10-11 nmbro <nmbro@users.noreply.github.com>

- drivers/apcmicrolink.c: fix(apcmicrolink): bump DRIVER_VERSION to 0.06 [#3728](#) already took 0.05 on master; this branch's driver changes need their own bump. Co-Author-By: Claude Opus 5.5 <noreply@anthropic.com>

2.25 2026-10-10 Jim Klimov <jimklimov+nut@gmail.com>

- docs/nut.dict: add timestamps too [#3669]

2.26 2026-10-09 Jakub Bartak <bartakjakub@gmail.com>

- NEWS.adoc, drivers/legrand-hid.c: drivers/legrand-hid.c: clear Unit/UnitExp on 1-bit items of Legrand devices Some Legrand firmware (seen on 1cb0:0035, which names itself "Reserve") leaves a Unit Exponent of -2 in effect for the 1-bit PresentStatus flags, so libhid scaled a set bit to 0.01: ACPresent never read as true (OB while on mains) and BelowRemainingCapacityLimit could never raise LB. Add legrand_fix_report_desc() which clears Unit and UnitExp on 1-bit items; it honors disable_fix_report_desc like the other fix-ups.

2.27 2026-10-08 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, drivers/apcmicrolink.c: fix(apcmicrolink): one record per serial poll The original serial driver read one record per upsdvr_updateinfo() call and published after each. The poll burst added for USB reads a full pass first, which on a 105-page serial unit takes ~6s, so a status change such as OL→OB could show up that much later, and instant commands waited as long. Serial polls continuously again, so the burst gains it nothing: take one record per call on serial. Also keep the overrun-while-auth-pending to USB, which is the only side with a pollinterval pause after a pass, and log the per-call burst summary at D4 when it is a single record. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- drivers/apcmicrolink.c: fix(apcmicrolink): frame traces to debug level 4 Every NEXT sent and every record received was traced at D2, which is about 95% of a D2 log and buries the session, auth and burst messages there and at D3. Move the TX ctrl and RX record traces to D4. TX write stays at D2: writes only happen for commands and auth. Suggested-by: Lukas Schmid <lukas.schmid@netcube.li> Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.28 2026-10-07 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, drivers/apcmicrolink.c: fix(apcmicrolink): no STOP on serial driver exit upsdvr_cleanup() sends STOP (0xF7) on the way out, so the device does not keep a half-walked session that the next client's INIT would land in. That was seen and fixed on the USB HID tunnel, but the original serial implementation never sent STOP on exit, and it now reaches serial too. Like the STOP before INIT, keep it to USB and leave the serial line exactly as the original serial driver did (issue #3587). Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- NEWS.adoc, drivers/apcmicrolink.c: fix(apcmicrolink): serial ready once auth is sent Startup used to count as complete as soon as the authentication response went out. For USB that was changed to keep polling until the device acknowledges it (up to MLINK_AUTH_GRACE_SEC, 15s), because the pollinterval cadence that followed startup never collected the acknowledgement. That wait applied to serial too, and while it runs a few consecutive 100ms read timeouts abort startup, where before the driver would already be in updateinfo and simply reconnect. Serial polls continuously again, so its regular polling collects the acknowledgement the way it always did. Restore the original readiness rule for serial; USB keeps the wait. When startup does fail after a successful handshake, name the unmet condition (no page 0, descriptor incomplete, auth not acknowledged) in the fatal message. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- NEWS.adoc, docs/man/apcmicrolink.txt, drivers/apcmicrolink.c: fix(apcmicrolink): poll serial continuously again The original serial driver forced poll_interval to 0 and polled the device back to back. Adding the USB transport (09737e779) switched both transports to the configured pollinterval, so a serial unit now sees a pause of pollinterval (2s by default) after every pass, a cadence it was never tested against. Restore continuous polling for serial only; USB keeps honoring pollinterval. The man page now notes that pollinterval does not apply on serial. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- NEWS.adoc, drivers/apcmicrolink.c: fix(apcmicrolink): serial INIT without STOP a990bd47a started sending STOP (0xF7) before every session INIT (0xFD), to match PowerChute's comms-recovery sequence on the USB HID tunnel. That path is shared with the serial transport, where the original implementation only ever sent INIT. A serial unit that had worked with INIT alone then never answered after switching to master: it received F7 FD and returned no data across all three startup attempts, and answered page 0 within about 62 ms once INIT went out alone again (issue #3587). Send STOP before INIT on USB only. The startup fatal message also always blamed missing standard HID Power Device usages, which is meaningless on serial, and did not say whether the handshake itself failed or the session started but never reached startup readiness. Say which

stage failed, and only mention the HID fallback on USB. Cause found and confirmed on serial hardware by Lukas Schmid, the original author of the serial driver. Reported-by: Lukas Schmid <lukas.schmid@netcube.li> Co-Author-By: Claude Opus 5.5 <noreply@anthropic.com>

2.29 2026-10-07 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/apcmicrolink.c: apply HID fallback only for USB builds [#3728]
- clients/upsset.c: restore comment for string length parsing logic [#3668]
- NEWS.adoc: regroup notes for `upsset.cgi` [#3668]
- NEWS.adoc: update upssched time width fix - refer to PR [#3669]
- NEWS.adoc: regroup the innovart33 entry [#3713]
- NEWS.adoc: for new vertiv-hid subdriver refer to PR [#3730]

2.30 2026-10-04 Warren Strong <warren.r.strong@gmail.com>

- NEWS.adoc, data/driver.list.in, docs/acknowledgements.txt, docs/man/usbhid-ups.txt, docs/nut.dict, drivers/Makefile.am, drivers/belkin-hid.c, drivers/usbhid-ups.c, drivers/vertiv-hid.c, drivers/vertiv-hid.h, scripts/upower/95-upower-hid.hwdb: drivers/vertiv-hid.c: add subdriver for Vertiv Liebert GXT5 (10af:1000) [#3613] The GXT5 shares the Liebert vendor ID handled by belkin-hid.c but only exposes a UPS.PowerSummary collection (no UPS.Input / UPS.Output / UPS.Battery), so only 8 of 83 Belkin/Liebert mappings matched: `ups.load` was never published and `battery.runtime` was served in minutes. Its firmware also declares the Volt items on reports 0x09 and 0x0b with Unit Exponent 5 instead of 7, so `battery.voltage` read 0.4 for a 40 V string. Add a `vertiv-hid` subdriver mapping the PowerSummary usages, converting runtime to seconds, and repairing the exponent via a `fix_report_desc()` hook (honours `disable_fix_report_desc`). The claim is strict on PID 0x1000 so it can never take a GXE3 / PST (10af:0002). `belkin_claim()` now rejects 10af:1000 explicitly, since with `productid` set in `ups.conf` it is consulted first and would otherwise claim the device; bump it to 0.25. Deliberately not mapped: `ConfigVoltage` (tracks mains, not battery nominal), `DesignCapacity` / `FullChargeCapacity` (`CapacityMode` is percent, not Ah), `ConfigApparentPower` / `ConfigActivePower` (Hz unit code, values are the mains frequency and likely a power factor). Tested on a GXT5 on armhf, mains only; on-battery transitions untested. Regenerated `scripts/upower/95-upower-hid.hwdb`; updated `driver.list.in`, `usbhid-ups` man page, NEWS.adoc, `nut.dict` and `acknowledgements.txt`. Claude Code (Claude Fable 5.1) assisted with the subdriver and analysis.

2.31 2026-10-03 d01 <d01.devel@gmail.com>

- NEWS.adoc, docs/man/apcmicrolink.txt, drivers/apcmicrolink.c: apcmicrolink: bump DRIVER_VERSION to 0.05, NEWS and man page for the HID fallback fix [issue #3587] Developed with AI assistance (Claude Code). Co-Author-By: Claude Fable 5.1 <noreply@anthropic.com>
- drivers/apcmicrolink.c: apcmicrolink: say so when startup fails because `hid_fallback` is disabled With "`hid_fallback = no`" a failed Microlink startup ended with "this device exposes no standard HID Power Device usages to fall back on", which is wrong for a device that has them: the fallback was switched off by configuration. Report that case as such. [issue #3587] Developed with AI assistance (Claude Code). Co-Author-By: Claude Fable 5.1 <noreply@anthropic.com>
- drivers/apcmicrolink-usb.c, drivers/apcmicrolink-usb.h: apcmicrolink: USB: HID fallback must not report status it never decoded The standard-HID fallback snapshot counted as "fresh" as soon as any of its reports arrived, including `RemainingCapacity` or `RunTimeToEmpty` alone. `ACPresent` and `Discharging` start out as 0, so until the first `PresentStatus` report was decoded the snapshot read "AC not present" and the driver published "OB" for a UPS that was on line power. Seen on an SMX1500 (firmware UPS 16.0) whose Microlink tunnel did not answer at boot: the driver started on fallback data, published OB, the snapshot then aged out and the data went stale. `upsmon` saw "on battery" followed by lost communications and forced a shut-down of a host that had mains power throughout. Track the decode time of `ACPresent` and of `Discharging` separately and only hand out a snapshot when both were decoded from a real report within the allowed age. Reports that carry only charge or runtime no longer refresh the status. Also clear the decoded values when the device is (re)opened, so nothing from before a reopen is published with a fresh status. Tested offline by feeding the decoder reports in the failing order (charge, runtime, then status): the old code yields OB after the first report, the new code yields nothing until the status report arrives, goes quiet again once the status is older than the limit even while charge reports keep coming, and still reports a real discharge. [issue #3587] Developed with AI assistance (Claude Code). Co-Author-By: Claude Fable 5.1 <noreply@anthropic.com>

2.32 2026-10-03 Ronny Brandt <ronnybrandt@gmail.com>

- NEWS.adoc, data/driver.list.in, docs/nut.dict, drivers/Makefile.am, drivers/libhid.c, drivers/libhid.h, drivers/microdowell-hid.c, drivers/microdowell-hid.h, drivers/usbhid-ups.c, scripts/upower/95-upower-hid.hwdb: usbhid-ups: add microdowell-hid sub-driver, fix max_report_size buffer sizing Adds support for the MicroDowell "Enterprise USB" (075d:0201), as fitted to the B.8 (800 VA). The device was unsupported by any subdriver; forcing it onto idowell-hid via vendorid/productid attaches but yields wrong data, because its report descriptor is broken in three independent ways. All three fixes are guarded by predicates that only match the broken encoding, so a corrected firmware revision is left untouched. 1) The descriptor emits PhysicalMinimum 0 / PhysicalMaximum 300 once and never resets it. Physical extents are global items, so every later Main item inherits them - including the single-bit PresentStatus flags. Their boolean 1 is then rescaled to 300 by logical_to_physical(), and since the info_lkp_t tables match on the exact value 1, no status flag is ever recognised: an online, charging UPS reports "OB OFF". 43 items were affected here. 2) In Input report 0x09 the nine PresentStatus usages are declared as a single 9-bit item carrying the ACPresent usage instead of nine 1-bit items, so ACPresent reads true whenever any flag is set. The Feature items for the same usages are declared correctly, so narrowing the Input item suffices. 3) UPS.PowerSummary.Voltage in report 0x08 is 16 bits wide but declares LogicalMaximum 255. GetValue() derives its mask from the logical range and discards everything above bit 7, turning 270 (27.0 V) into 14 (1.4 V). Separately, this firmware only answers a GET_REPORT when asked for more bytes than it declared; asking for the declared 4 to 6 bytes returns all zeroes. The subdriver therefore enables the existing max_report_size tweak - which turned out not to work as intended: drivers/libhid.c:174 r = max_report_size ? sizeof(rbuf->data[id]) : ... data[] is an array of pointers, so sizeof() yields the size of a pointer (8 bytes on LP64, 4 on ILP32) rather than the size of the buffer. The tweak therefore never asked for the full buffer. Worse, the buffer is allocated as calloc(rbuf->len[id], 1), so for any report shorter than a pointer the device is invited to write past the allocation - observed here as refresh_report_buffer: expected 4 bytes, but got 8 instead on reports 0x07, 0x0a and 0x0b. This adds an explicit cap[] alongside len[], set in new_report_buffer() right where the buffer is allocated, and asks for cap[] unconditionally: it equals len[] unless the tweak requested more, so the old ternary was redundant. Note for review: this changes the effective request size for devices already using the tweak (some APC Back-UPS models) from a pointer's width to MAX_REPORT_ALLOC. The constant is a placeholder - please pick whatever value you consider safe for those devices. I have no APC hardware to test against. Verified on the hardware with a driver built from this branch. Pulling the mains lead for 60 s gives clean transitions in both directions: t+32.7s Charging 1 → 0 t+32.7s Discharging 0 → 1 t+34.7s ACPresent 1 → 0 ups.status carries OB from here t+94.7s ACPresent 0 → 1 ups.status back to OL t+96.7s Charging 0 → 1 t+96.7s Discharging 1 → 0 OB was present for exactly the 58 s the device reported no mains, and battery.charge, battery.voltage, battery.voltage.nominal, battery.type, battery.charge.low/warning and the nominal ratings all match values decoded independently from /dev/hidraw0. NEWS.adoc carries one entry for the new subdriver and one for the libhid fix, with the technical terms they introduce added to docs/nut.dict so the docs spellcheck stays green. data/driver.list.in gains the USB interface of the Enterprise B.8; note the existing "Enterprise B8" entry there claims support level 5 via the serial microdowell driver, but the RS232 port of this unit transmits nothing at all (see #3712), so the USB entry is listed separately and conservatively at level 2. Not mapped on purpose, both documented in the source: UPS.PowerSummary. PercentLoad is declared but stayed 0 throughout that battery run with a real load, and RunTimeToEmpty always reads 0xFFFF ("unknown"). Output voltage is also left out - the declared Output.Voltage field is constant on this firmware, while an undocumented field one byte earlier does track the line. Co-Authored-By: Claude Opus 5 (1M context) <noreply@anthropic.com>

2.33 2026-10-02 Michael Walker <34684351+walkermc20@users.noreply.github.com>

- NEWS.adoc, drivers/idowell-hid.c: idowell-hid: fix Usage Page of GoldenMate Good/ShutdownImminent flags GoldenMate LiFePO4 packs (075d:0300, 06da:ffff) declare seven 1-bit PresentStatus flags in Feature report 0x01, but select Usage Page 0x85 (Battery System) for ACPresent and never switch back to 0x84 (Power Device). Usages 0x61 and 0x69 are thus decoded as AtRateTimeToEmpty and AverageTimeToEmpty (run-time counters, meaningless as single bits) instead of the intended Good and ShutdownImminent, and get ignored. Rewrite just those two usages in fix_report_desc so the existing PresentStatus.Good / PresentStatus.ShutdownImminent mappings apply. The fix only touches 1-bit Feature items directly under PresentStatus in report 0x01, so a corrected firmware is left alone. Verified against the issue's 291-byte descriptor with NUT's HID parser (only those two items change; a second pass is a no-op), and on a live 06da:ffff unit: Good=1 and ShutdownImminent=0 both on mains and on battery while ACPresent/Discharging flip as expected. Refs: <https://github.com/networkupstools/nut/-issues/3015> Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.34 2026-10-02 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: rearrange numerous blocks about apcmicrolink driver into one [#3714]

- NEWS.adoc, UPGRADING.adoc: mark entries fixed by [PR #3720](#)
- NEWS.adoc, docs/macros.txt: settle on "-ize" as the spelling consistently acceptable to checkers on different platforms [[#3674](#), [#3678](#), [#3680](#)]
- Makefile.am: for rules often executed interactively, report SUCCESS to ease maintainer work
- AGENTS.md: clarify maintenance of NEWS.adoc and UPGRADING.adoc
- docs/nut.dict: Add new terms [[#3726](#)]

2.35 2026-09-29 TomZanna <git@tomzanna.com>

- docs/man/upscmd.txt: docs: Document optional value argument for upscmd Add the optional *value* argument to the upscmd man page synopsis and describe both the *command* and *value* parameters.

2.36 2026-09-28 user01010111 <lapses.50.booster@icloud.com>

- m4/nut_check_semaphores.m4: m4: preserve detected semaphore libraries in cache Append the optional librt fallback to nut_cv_SEMLIBS so detected semaphore libraries survive the cache assignment.

2.37 2026-09-27 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, docs/apcmicrolink-descriptors.txt, drivers/apcmicrolink-maps.c: feat(apcmicrolink): read the object-A calibration status The driver maps the runtime calibration status at 2:13 and 2:4.5.13, but a Smart-UPS 500 (SCL500RM1UC) has neither: of the 242 usages in its descriptor, the only calibration status is A:6.13, which the protocol spec lists as RunTimeCalibrationStatus_BF (FQUN 2:4.A:6.13). That model therefore published no calibration state at all, and a calibration in progress never raised CAL. Map A:6.13 read-only with apc_calibration_status_map, and add it to microlink_desc_publish_map with the CAL-only map, so a running calibration is reported in ups.status the same way as for the other two registers and a latched outcome cannot leak there. It gets its own name, experimental.microlink.ups.calibration.status, rather than sharing one of the two *.calibration.result names. Object A has its own attribute namespace, and nothing yet shows that it carries the same value as 2:13 or 2:4.5.13 on a device that has those as well; mapping two usages onto one variable would make the published value depend on descriptor order (docs/apcmicrolink-descriptors.txt, "Two scopes, one NUT name"). The descriptor notes now record A:6.13 as the first identified usage of object A. Tested on the SCL500RM1UC: the variable is published ("None", register 0x0000) and ups.status stays "OL". A calibration could not be started on this unit to see CAL raised: it has no calibration command register. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>
- NEWS.adoc, docs/man/apcmicrolink.txt, drivers/apcmicrolink.c: fix(apcmicrolink): only advertise commands the device can take upsdv_initinfo() registered all nine simple instant commands (test.battery., **test.panel.start**, **beeper.**, calibrate., **bypass.**) unconditionally, but each of them writes one descriptor usage, and microlink_send_descriptor_mask_value() refuses - silently - to write a usage the device's descriptor does not contain. On a model without the register the command is listed by upscmd -l yet can only ever fail. Seen on a Smart-UPS 500 (SCL500RM1UC) over USB: its descriptor has 242 usages and none of them is 2:12, so calibrate.start was advertised and then rejected with nothing but "instcmd: FAILED command [calibrate.start]" in the log, at any debug level. Move the command-to-usage mapping into one table, microlink_simple_cmd_paths[], that both the instcmd handler and a new microlink_register_simple_commands() use, and only register a command when its usage is present and writable under the same conditions the write path checks. Registration follows microlink_register_outlet_commands(): it waits for the descriptor, so it is called at init and again from upsdv_updateinfo() for a USB start that begins on the HID fallback, and it is reset with the other session state. Commands left out are logged at debug level 1 with the missing usage. Tested on the SCL500RM1UC (clean startup): calibrate.start and calibrate.stop are no longer offered and the debug log names 2:12 as missing; the other seven simple commands and all outlet, load and shutdown commands are still offered. The late-registration path after a HID-fallback start was not exercised. The man page command list is brought in line with the driver while at it: it named test.panel.stop, which has never been implemented, and omitted calibrate.* and bypass.*. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

- NEWS.adoc, drivers/apcmicrolink-maps.c: fix(apcmicrolink): CAL from 2:4.5.13 too, name the InProgress bit Follow-up to "apcmicrolink: keep the calibration result out of ups.status" (PR #3714), which publishes only CAL from the calibration status register 2:13. The driver also maps the UPS-scope spelling of that usage, 2:4.5.13 (as experimental.ups.calibration.result), and some devices populate that one instead - the same reason 2:11 and 2:4.5.11 are both handled for the test result. Add it to microlink_desc_publish_map with the same CAL-only map, so a running calibration reaches ups.status on those devices too. Also spell the bit as APC_CAL_STATUS_IN_PROGRESS: apc_common.h already defines it next to apc_calibration_status_map, and nothing used it. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Opus 5.5 <noreply@anthropic.com>

2.38 2026-09-27 Arcadiy Ivanov <arcadiy@ivanov.biz>

- NEWS.adoc, UPGRADING.adoc, docs/man/nutdrv_qx.txt, drivers/Makefile.am, drivers/nutdrv_qx.c, tests/.gitignore, tests/Makefile.am, tests/nutdrv_qx_cypress_ustest.c: nutdrv_qx: keep cypress 0665:5161 replies in step with commands A Cypress 0665:5161 USB-serial bridge keeps only the newest 8 input reports (64 bytes) while nobody reads its interrupt endpoint, and drops the oldest ones. After a reply was left unread (e.g. the UPS answered only after the driver stopped waiting) or the UPS sent data unasked, the next command read that stale data (for replies longer than 64 bytes, also truncated) as its own reply, and every later reply was one command behind. With NUT 2.8.5 this was seen twice in the field on an OptiUPS (Voltronic protocol, 76-byte QGS reply): every QGS reply lacked its first 16 bytes and the driver silently reported stale data until it was restarted. usbmon showed that "reply" arriving 8-15 ms after the command, i.e. already queued, where real replies take 80-408 ms. With the cypress_drain_quirk enabled, cypress_command() now: * tracks whether a reply is still outstanding; then, or once stale data turns up, it drains until the endpoint stays quiet for 500 ms, longer than the pauses the UPS makes within a reply (up to 160 ms were seen); * stops draining after 32 reports or 3 seconds and returns the new QX_USB_ERROR_DESYNC, on which qx_command() resets the USB device and reconnects, up to 3 times in a row before exiting, instead of exiting at once with a misleading "Got disconnected by another driver"; * discards zero-length reports while draining, instead of returning them as an empty reply without sending the command; * logs discarded stale data once at NOTICE level (naming the readable part of the command), not on every poll. The quirk default (on for 0665:5161) is now decided once, after the device is opened, so it also applies with an explicit "subdriver = cypress", which used to leave it off; cypress_0665_5161_subdriver() is folded into cypress_subdriver(). qx_command() no longer reports a reconnection as successful when the device was just closed again for another reconnection, so such attempts now count towards a positive reconnect_max_tries. Add tests/nutdrv_qx_cypress_ustest.c: it includes nutdrv_qx.c and runs it against a simulated bridge modelled on the usbmon measurements (queue depth, reply latency, pauses within replies), substituting the libusb, clock and log calls. It is built like the driver (with serial support when configured), and runs its 18 cases in forked children where available. AI assistance: Claude Code with Claude Opus 5.5 was used for the investigation, implementation, tests and validation. The human contributor remains responsible for the change.

2.39 2026-09-26 d01 <d01.devel@gmail.com>

- NEWS.adoc, drivers/apcmicrolink.c: apcmicrolink: bump DRIVER_VERSION to 0.04, NEWS entry for the SMX1500 fixes Co-Authored-By: Claude Fable 5.1 <noreply@anthropic.com> Claude-Session: https://claude.ai/code/session_01MF4kawLRUbridaA
- drivers/apcmicrolink-usb.c, drivers/apcmicrolink-usb.h, drivers/apcmicrolink.c: apcmicrolink: USB: keep the report's zero padding away from the frame scanner Every tunnel Input report carries one Microlink record followed by 0x00 padding up to the report's fixed size (observed on a Smart-UPS X 1500, FW "UPS 16.0": a 35-byte record plus 28 zero bytes, in 13,040 of 13,040 reports). The frame scanner works on a byte stream and takes the first offset at which the checksum validates. Leading zero bytes are invisible to the Fletcher-mod-255 checksum, so a record whose last check byte is 0xFF (and whose byte 32 is non-zero) also validates one byte early. The scanner then takes it for a page 0, the guard from f0a4525 rightly rejects that page 0 ("ignoring implausible page 0 i. width 0"), and the real record is lost for that pass. On the test unit this hit the pages with live counters two or three times a minute. The USB transport knows where each report ends, so let it settle it: right after the parser has extracted a frame, drop the rest of the current report if and only if it is all zeros. A non-zero remainder is left to the scanner as before, so nothing that could be data is ever discarded (a complete record cannot hide in all-zero bytes, since a computed check byte is never 0x00), and a record wider than one report still accumulates across reports. Verified offline by replaying a captured session through a model of the scanner (the unpatched model reproduces the driver's log exactly; the patched one yields zero bogus page 0s and zero lost records, output byte-identical to the wire), and live over 180 s with usbmon (12 alias opportunities on the wire, 0 misparses, 6,451 parsed records identical to the 6,451 reports

received). The per-record "skipped 28 stray byte(s)" debug line goes away with it. [issue #3587] Developed with AI assistance (Claude Code); reviewed and tested by the submitter on the hardware above. Co-Authored-By: Claude Fable 5.1 <noreply@anthropic.com>

- drivers/apcmicrolink-maps.c: apcmicrolink: keep the calibration result out of ups.status microlink_desc_publish_map fed the whole apc_calibration_status_map for descriptor 2:13 to status_set(), so the labels of the calibration RESULT register (Passed, Failed, Refused, ...) leaked into ups.status as bogus status tokens. A Smart-UPS X 1500 (FW "UPS 16.0") whose last runtime calibration had ended "Failed" reported "OL Failed" permanently, and upsmon logged "unclassified status token: [Failed]". Only a calibration in progress belongs in ups.status: map bit 1 (InProgress) to CAL and nothing else. The result itself is still published as experimental.microlink.ups.calibration.result. [issue #3587] Developed with AI assistance (Claude Code); reviewed and tested by the submitter on the hardware above. Co-Authored-By: Claude Fable 5.1 <noreply@anthropic.com>

2.40 2026-09-25 Viktor Drobot <linux776@gmail.com>

- NEWS.adoc, drivers/nutdrv_qx_innovart33.c: Try to fix #3661

2.41 2026-09-25 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: move LISTEN directive updates to upsd block [#3709]

2.42 2026-09-24 user01010111 <lapses.50.booster@icloud.com>

- NEWS.adoc: docs: reference PR #3709 in LISTEN release note

2.43 2026-09-23 user01010111 <lapses.50.booster@icloud.com>

- NEWS.adoc, common/nutconf.cpp, common/str.c, conf/upsd.conf.sample, docs/man/upsd.conf.txt, include/nutconf.hpp, include/str.h, scripts/augeas/nutupsdconf.aug.in, scripts/augeas/tests/test_nut.aug, server/conf.c, tests/nutconf_parser_ut.cpp, tests/test_upsdconf.c: upsd: accept inline ports in LISTEN directives

2.44 2026-09-23 Jordan Rodgers <com6056@gmail.com>

- NEWS.adoc, tests/NIT/nit.sh, tools/nut-scanner/nutscan-init.c, tools/nut-scanner/scan_avahi.c, tools/nut-scanner/scan_ipmi.c, tools/nut-scanner/scan_nut.c, tools/nut-scanner/scan_snmp.c, tools/nut-scanner/scan_upower.c, tools/nut-scanner/scan_usb.c, tools/nut-scanner/scan_xml_http.c: nut-scanner: one load verdict per library at the default debug level [#3654] Each nutscan_load_*_li printed "Cannot load X library (name) : reason. X search disabled." at debug level 0 for every failed attempt. Now that nutscan_init() really tries each candidate name in turn, that made a missing library print one line per name, and a library that loads on a later name still claimed its search was disabled. Move the per-attempt messages, and the loaders' "library not found" messages, to debug level 1, where the name tried and the reason stay available, and drop their "X search disabled" suffix since the scan goes on to the next name. nutscan_init() now prints a single line per library that could not be loaded at all. A library that loads keeps its existing debug level 1 message, so a normal run prints nothing new. tests/NIT/nit.sh excused a missing libupsclient by matching the old per-attempt wording, so point that pattern at the new summary line.

2.45 2026-09-23 Jim Klimov <jimklimov+nut@gmail.com>

- server/netssl.c: quiesce unreachable-code pragmas [#3329] Some macros in OpenSSL on certain platforms resolve to always-true at compile time, so embedded conditionals are pointless. Not much we can do about them though.
 - UPGRADING.adoc: move PLANNED placeholder back up, attribute Perl fixes to [PR #3634]
 - NEWS.adoc: refactored NUT_CHECK_SEMAPHORES into a separate m4 file [#3675]
 - NEWS.adoc: updated m4 macro docs [#3674]
-

2.46 2026-09-22 user01010111 <lapses.50.booster@icloud.com>

- docs/config-notes.txt, docs/man/upsmon.conf.txt, docs/man/upsmon.txt: docs: clarify Windows NOTIFYCMD quoting Distinguish configuration tokens from Windows child arguments and document the tested data-path quoting form and notification-message limitation. OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.

2.47 2026-09-21 Jim Klimov <jimklimov+nut@gmail.com>

- configure.ac: lower-case NUT_REPORT_FEATURE for WITH_THREADING to match other final reports [#3674]
- docs/macros.txt: revive and rephrase the rationale behind our multiple macro files [#3674]
- docs/macros.txt: rephrase explanation of NUT_REPORT_FEATURE [#3674]
- m4/nut_check_semaphores.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108, #3675, #1518]
- m4/nut_check_semaphores.m4: add gate against re-runs [#3675]
- docs/man/nutscan_scan_snmp.txt: rephrase description about security levels [#3680]
- docs/config-notes.txt: comment about the two-token example with multiple escaped spaces [#3685]
- docs/man/upsmon.txt: revise how "message" is the first arg after the whole NOTIFYCMD definition [#3685]
- conf/upssched.conf.sample.in, docs/man/upssched.conf.txt: revise how "timername" is the first arg after the whole CMDSCRIPT definition [#3685]
- conf/upssched.conf.sample.in: revise "Note 1" for PIPEFN [#3685]
- docs/nut.dict: add nullptr [#3678]

2.48 2026-09-20 user01010111 <lapses.50.booster@icloud.com>

- docs/man/upsmon.txt: upsmon: restore direct TLS configuration guidance Retain the supported upsmon certificate directives, use CERTIDENT, and mention AUTHCONF for selecting an additional authentication file. OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - conf/upssched.conf.sample.in, docs/man/upssched.conf.txt: upssched: clarify timer argument ordering Explain the first-and-only argument case and how configured CMDSCRIPT arguments precede the timer name. Use neutral wording in the individual timer descriptions. OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3694](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3695](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3699](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3697](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3696](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3693](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3698](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3692](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3686](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3687](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3689](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
-

- NEWS.adoc: NEWS: reference [PR #3690](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3691](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3688](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3685](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3679](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3683](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3680](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3684](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3682](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3681](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3677](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc: NEWS: reference [PR #3678](#) OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/nutdrv_qx.txt, docs/nut.dict: Add required USB identity fields to the Gtec example OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/powervar_cx_ser.txt, docs/man/powervar_cx_usb.txt: Remove unsupported Powervar quick-test configurability guidance OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/developers.txt, docs/man/apcupsd-ups.txt, docs/man/hosts.conf.txt, docs/man/libupsclient-config.txt, docs/-man/nutauth.conf.txt, docs/man/oneac.txt, docs/man/upscli_dump_authconf_item.txt, docs/man/upscli_get_or_create_ssl_context.txt, docs/man/upsclient.txt, docs/man/upsset.conf.txt, docs/nut.dict, docs/outlets.txt, scripts/augeas/README.adoc, scripts/external_apis/enphase/README.adoc, scripts/external_apis/enphase/enphase-monitor.in, scripts/usb_resetter/README.adoc: Correct documentation references and exact command/configuration names OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/upsdrvsvctln.txt, scripts/upsdrvsvctln/upsdrvsvctln.in: Correct the upsdrvsvctln command reference OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/mge-shut.txt: Remove the ignored mge-shut notification workaround OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/generic_gpio.txt: Correct GPIO driver executable and udev examples OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/nut.dict, scripts/fuse/README.adoc: Correct the FUSE data-walk example OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/belkinunv.txt, docs/man/genericups.txt, docs/nut.dict: Correct standalone belkinunv and genericups command examples OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/upsd.txt: Correct upsd state-path precedence and debug-output documentation OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/apc_modbus.txt, drivers/apc_modbus.c: Update APC Modbus capability, timeout and attempt-limit documentation OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/apcmicrolink.txt, docs/man/apcsmart.txt: Qualify APC Microlink support and USB detachment guidance OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, scripts/python/module/PyNUT.py.in, scripts/python/module/README.adoc: Correct PyNUT command and return-type examples OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
-

- NEWS.adoc, docs/man/upsc.txt, docs/man/upscmd.txt, docs/man/upslog.txt, docs/man/upsmon.txt, docs/man/upsrw.txt: Correct client connection defaults and remote query example OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/upscmd.txt, docs/man/upslog.txt, docs/man/upsmon.txt, docs/man/upsrw.txt, docs/man/upsstats.cgi.txt: Clarify client authentication configuration and unattended commands OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, conf/upssched.conf.sample.in, docs/config-notes.txt, docs/man/upsmon.conf.txt, docs/man/upsmon.txt, docs/man/upssched.conf.txt: Correct notification arguments, shared timers and FINALDELAY documentation OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, UPGRADING.adoc, docs/man/upscli_readline.txt, docs/man/upscli_sendline.txt, docs/man/upscli_splitaddr.txt, docs/man/upscli_splitname.txt, docs/man/upscli_ssl_caps.txt, docs/man/upscli_upslog_set_debug_level.txt, docs/nut.dict: Correct upsclient API declarations and return-value descriptions OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/developer-guide.txt, docs/man/nut-recorder.txt: Correct the recording and device-simulation walkthrough OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/nutscan_add_device_to_device.txt, docs/man/nutscan_add_option_to_device.txt, docs/man/nutscan_free_dev, docs/man/nutscan_get_serial_ports_list.txt, docs/man/nutscan_new_device.txt, docs/man/nutscan_scan_eaton_serial.txt, docs/man/nutscan_scan_ipmi.txt, docs/man/nutscan_scan_snmp.txt, docs/man/nutscan_upslog_set_debug_level.txt, tools/nut-scanner/RE: Correct scanner-library API documentation and consumer example OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/man/nut-scanner.txt, docs/man/nutscan_scan_nut_simulation.txt, tools/nut-scanner/nut-scanner.c: Correct scanner simulation options and scan examples OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/design.txt, docs/net-protocol.txt, docs/sock-protocol.txt: Correct protocol notation and tracking descriptions OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/configure.txt, docs/developers.txt, docs/download.txt: Correct source retrieval and configure documentation OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, docs/new-clients.txt: Initialise the connection pointer in the C++ client example OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - INSTALL.nut.adoc, NEWS.adoc: Make the installation staging example preserve the source tree OpenAI Codex (gpt-6-astra, xhigh) assisted with preparation and validation.
 - NEWS.adoc, common/nutstream.cpp, tests/nutstream_ut.cpp: libnutconf: fix character reads from Windows sockets
 - tests/Makefile.am, tests/upssched-list-timers-test.c: tests: fix upssched regression linkage and fixture lifetime Link the version archive before the common library so Windows builds with shared private libraries resolve its dependencies. Give the timer fixture static storage and explain the intentional implementation inclusion used to exercise the private handler and real formatter. Reproduced the reported shared-library link failure on native MinGW64; the corrected target and existing static target pass all four cases. Linux static and bundled-formatter checks and distcheck-light pass. AI assistance: OpenAI Codex was used to prepare and validate this change.
 - configure.ac, docs/macros.txt, docs/nut.dict, m4/nut_check_semaphores.m4: configure: move semaphore checks into an M4 macro Move the semaphore probes into NUT_CHECK_SEMAPHORES without changing their body or execution order. Document the macro and add SEMLIBS to the spelling dictionary. Related to [#1518](#).
 - docs/macros.txt, docs/nut.dict: docs: refresh the M4 macro reference
 - NEWS.adoc, docs/man/hwmon_ina219.txt, docs/man/liebert.txt: docs: document Liebert and INA219 driver options I documented the existing mfr and model string options and their defaults for liebert, whose manual previously said it had no extra settings. For hwmon_ina219, I documented sysfs_dir and its port=auto condition, corrected the hwmon directory path, and added the integer range and default for the low battery threshold. I also added a NEWS entry. I checked each description against the option registration and consuming code. The affected man pages and HTML pages build through the project recipes and pass the manual checks. ASCII and whitespace checks, and make distcheck-light, passed.
-

Full `make spellcheck` passed with a local dictionary supplement for `kernel's` and `firmware's`, which also fail on unchanged upstream with the installed Aspell dictionary; the project dictionary is unchanged. Related to [#2422](#). This covers the two driver manuals and leaves the wider option audit open. I prepared this change with assistance from OpenAI Codex.

- NEWS.adoc, clients/upssched.c, tests/Makefile.am, tests/upssched-list-timers-test.c: upssched: preserve full-width expiry timestamps in timer listings `upssched -l` narrows each timer's `time_t` expiry to `long` before printing it. On systems with 64-bit `time_t` and 32-bit `long`, dates beyond January 2038 can be truncated or displayed as negative timestamps. Use the existing `intmax_t` formatting convention and add a regression that checks the actual `LIST-TIMERS` response around the 2038 boundary and in 2100. Related to [#3096](#). Validated with native Windows/MinGW64 failing-before/passing-after checks and the full CLI using disposable pipes, including a short timer and cancellation. Linux normal and forced-fallback checks, the configured test suite and `distcheck-light` passed. The changed release notes render correctly through the project's HTML target. Spellchecks used a temporary dictionary containing two pre-existing rejected possessives; the project dictionary is unchanged. AI assistance: OpenAI Codex was used to prepare the change and tests.

2.49 2026-09-20 user01010111 <12504630+user01010111@users.noreply.github.com>

- NEWS.adoc, docs/man/nutupsdrv.txt: docs: explain direct driver commands with `sdcommands` I added a `nutupsdrv(8)` example for cancelling a pending shutdown through the existing `sdcommands` facility, with a cross-reference to `ups.conf(5)`. The text explains command-line precedence, fallback ordering and why the driver daemon must be stopped for this override. The example uses `-F` to keep the driver in the foreground. [#2686](#) introduced the functionality requested in [#1677](#); this change documents its use from the command line. I verified the configured command override, reversed option ordering, unknown-command failure and fallback ordering against an isolated serial simulator using a fresh build of `a66c0090`. I built the man page, HTML manual and release notes, and ran `make distcheck-light` with the dummy-`ups` driver selection. The distribution check passed with a temporary Aspell supplement for existing `kernel's` and `firmware's` spelling failures; the repository dictionary is unchanged. AI assistance: OpenAI Codex, using `gpt-6-astra` with `xhigh` reasoning effort.

2.50 2026-09-20 Jordan Rodgers <com6056@gmail.com>

- NEWS.adoc, tools/nut-scanner/nutscan-init.c: tools/nut-scanner/nutscan-init.c: unload before every fallback candidate [\[#3654\]](#) The earlier commit only reset the module handle between the `SONAME_LIB*` and `SOFIE_LIB*` attempts. Every other candidate in the `lt_dlopen()` fallback chains has the same problem, since a failed attempt is remembered in the handle and `nutscan_load_*_library()` then short-circuits on that memory instead of really trying the remaining names. Unload before each of them too, including the `libupsclient` chain which has no `SONAME_LIB*` of its own. The reason is spelled out once per block in the comment above each chain, replacing the per-site notes the earlier commit added. Note that the candidates after the first two were never really tried before this, so a library which is absent entirely now reports every name it tried at debug level 0 instead of just those.

2.51 2026-09-20 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/Makefile.am: EXTRA_DIST and spellcheck scripts/fuse files [\[#3698, #2591\]](#)
- docs/new-clients.txt: fix `C+\+` pointer init from `NULL` to `nullptr` [\[#3678\]](#)
- docs/man/nut-scanner.txt: restore the USB scan example, and comment about enumeration-dependent values [\[#3684\]](#)
- docs/design.txt: restore the note about data server round-robinning and quick replies [\[#3582\]](#)
- docs/developers.txt: revise suggestions about Git checkout [\[#3681\]](#)
- UPGRADING.adoc, docs/man/Makefile.am, docs/man/{generic_gpio.txt ⇒ generic_gpio_libgpiod.txt}, scripts/obs/debian.nut-gpio.manpages: docs/man/Makefile.am, UPGRADING.adoc: rename man page of `generic_gpio_libgpiod` driver from `generic_gpio` it was since draft [\[#3693, #1855\]](#)
- scripts/fuse/README.adoc: suggest a more idiomatic way of grepping for non-trivial contents [\[#3698, #2591\]](#)
- NEWS.adoc, scripts/Makefile.am: scripts/Makefile.am: EXTRA_DIST and spellcheck scripts/fuse files [\[#3698, #2591\]](#)

2.52 2026-09-19 user01010111 <lapses.50.booster@icloud.com>

- NEWS.adoc, clients/upsset.c, tests/Makefile.am, tests/upsset-string-test.py: upsset: handle string lengths above 127 Keep string lengths as long through checked metadata parsing and HTML rendering. Report invalid or non-positive lengths as an unknown type instead of asserting or narrowing them to char. Add a real-CGI loopback regression covering length boundaries, malformed metadata, a 128-character save/readback, encoding and the submission cap. Register the test conditionally and document the fix in NEWS. Fixes [#3545](#). AI assistance: OpenAI Codex, using gpt-6-astra for independent review, implementation and validation.

2.53 2026-09-19 Jim Klimov <jimklimov+nut@gmail.com>

- docs/config-prereqs.txt: mention jq for NUT CI Swarm style agents

2.54 2026-09-18 antuspcm <18096106+antuspcm@users.noreply.github.com>

- data/driver.list.in, docs/man/usbhid-ups.txt: document Phoenixtec devices that require the pollonly flag The ION F11 (USB ID 06da:ffff) is already claimed by liebert-hid, but only answers Feature reports and never sends Interrupt In reports. Without pollonly the driver enumerates all 45 HID objects, then logs "Got 0 HID objects" every poll with "nut_libusb_get_interrupt: Connection timed out" and publishes nothing, which looks like an unsupported device rather than a missing flag. With pollonly, upsc reports ups.status, battery.charge, battery.runtime, battery.voltage, input/output voltage and frequency, ups.load, beeper and shutdown delays. Verified on an F11-650. Tested on FreeBSD 15.1 (OPNsense 26.7), NUT 2.8.5, subdriver "Phoenixtec/Liebert HID 0.41".

2.55 2026-09-18 Jim Klimov <jimklimov+nut@gmail.com>

- server/netssl.c: not all OpenSSL versions defined X509_V_ERR_HOSTNAME_MISMATCH [[#3640](#), [#3331](#)]
- NEWS.adoc: update PR number for dummy-ups stale support [[#3648](#)]
- NEWS.adoc: move santak subdriver to nutdrv_qx section [[#3662](#)]
- tools/nut-scanner/nutscan-init.c: squash diagnostic pragmas using header files [[#3654](#), [#3650](#)]

2.56 2026-09-17 lfntn <6958805+lfntn@users.noreply.github.com>

- drivers/nutdrv_qx_voltronic-qs.c: nutdrv_qx: fix -DTESTING build broken by a missing comma voltronic_qs_testing[] is missing the comma separating two initialisers, so any build with -DTESTING fails: nutdrv_qx_voltronic-qs.c:141:9: error: expected } before { token nutdrv_qx_voltronic-qs.c:137:50: note: to match this { This blocks the per-subdriver testing_t tables for the whole nutdrv_qx family, not just voltronic-qs, since they all build into the same binary. Verified before and after on NUT master: the -DTESTING build fails as above without this change, and with it the driver builds and selects "Voltronic-QS 0.11" against its own canned frames. Fixes: <https://github.com/networkupstools/nut/issues/3660> Co-Authored-By: Claude Opus 5 <noreply@anthropic.com>
- NEWS.adoc, data/driver.list.in, docs/man/nutdrv_qx.txt, docs/nut.dict, drivers/Makefile.am, drivers/nutdrv_qx.c, drivers/nutdrv_qx_santak.c, drivers/nutdrv_qx_santak.h: nutdrv_qx: add santak subdriver for single-phase Castle C*K units Santak Castle C*K UPSes speak the same Q6 dialect as innovart31, but are single-phase and NAK the F (ratings) query, so innovart31 rejects them with "Device not supported!" and the generic q6 subdriver reports three bogus phases with L1-N qualified variable names. This adds a santak subdriver, developed and tested against a Castle C1K reporting model C1K(G7) over RS232 at 2400 8N1: - Q6 → input/output voltage and frequency, load, pack-level battery.voltage, temperature, battery.runtime, battery.charge - WA → output current, real power, apparent power - BPS → bypass voltage and frequency - WH → ups.model, nominal output voltage/frequency and the input transfer window; this replaces the ratings that F would normally provide, since these units NAK both F and I - FW? → ups.firmware - SASV07? → ups.serial - Q1 → status bits only (its battery.voltage is per cell; Q6 gives the pack voltage directly, so battery_voltage_reports_one_pack must NOT be set for these devices) The subdriver is listed ahead of the generic q2/q6 probes because those would otherwise claim the device; claim() requires Q6, Q1, WA, BPS, WH, FW? and SASV07?, and additionally verifies the single-phase signature (the "---" placeholder in the Q6 input L2/L3

slots) so that three-phase Innova units are left to their own subdrivers. That rejection path was verified by feeding innovart31's test frames through `santak_claim()`. Commands confirmed NAKed by this device and therefore not mapped: I, F, Q2, Q3, Q5, QS, QMOD, QFLAG, QGS, QMD, QID, QRI, QBV, QBYV, QBYF, QPI, QVFW, QDI, QBT, QT, QMF, QENF, QFET, QOPF, QFRE, QCHT, QL DL, SS? and BP?. Q4 and SASV08? do answer but their contents could not be identified without vendor documentation, so they are recorded in a comment rather than guessed at. Only the Castle C*K HCL entry is moved to `nutdrv_qx`; the MT*-PRO line from the same issue is left alone as no such device was available to test. Also bumps `nutdrv_qx DRIVER_VERSION` to 0.56, and adds "Santak", "santak" and "WH" to `docs/nut.dict` so that `spellcheck` passes. Refs: <https://github.com/networkupstools/nut/issues/1039> Co-Authored-By: Claude Opus 5 <noreply@anthropic.com>

2.57 2026-09-17 Jordan Rodgers <com6056@gmail.com>

- `tools/nut-scanner/nutscan-init.c`: really try `SOFILIB*` after a failed `SONAME_LIB*` attempt [#3654] Each `nutscan_load_*_library` caches a failure by setting `dl_handle` to `(lt_dlhandle)1`, and returns 0 for every later call until the module is unloaded. In the `lt_dlopen()` fallback branch, that made the `SOFILIB*` candidate unreachable once the `SONAME_LIB*` attempt before it failed, so a library reachable only under its versioned file name (a bundle with no `SONAME` symlink on the run-time linker search path) stopped being found. Unload the failed attempt first, which resets the handle without an extra `lt_dlexit()`, so the next candidate is really tried. Note that a library which is absent entirely now reports two failed candidates at debug level 0 instead of one, since the second name is really tried.

2.58 2026-09-17 Jim Klimov <jimklimov+nut@gmail.com>

- `tools/nut-scanner/nutscan-init.c`: add pragmas against "-Wunreachable-code" for `strcmp(SONAME_LIBX, SOFILE_LIBX)` [#3654] Newer builders (notably those with clang-21 and clang-22) on some agents correctly identify that we compare identical (macro) strings and the clause would never run. Ideally the comparison would be pre-processor equality check for strings, but as that is missing (or not portable), we compare to avoid work in vain at run-time - and now quiesce the compiler warnings about that on systems that do benefit from this check. To be revised after PR #3650 is merged, to simplify these pragma blocks.

2.59 2026-09-16 Priyanshu <ankurdwivedi75@gmail.com>

- `drivers/nutdrv_qx_voltronic.c`: `nutdrv_qx`: add PI71 protocol support for Microtek/Voltronic UPS - Allow digit 7 in protocol string validation - Add case 71 to accepted protocols switch - Handle -- (ups.type) and - (status bits) in `voltronic_status()` Tested with Microtek GI3KL 3000VA online UPS (0665:5161, PI71)

2.60 2026-09-16 Jordan Rodgers <com6056@gmail.com>

- `NEWS.adoc`, `tools/nut-scanner/nutscan-init.c`: `nut-scanner`: record the `libupsclient` fallback result in `nutscan_avail_nut` When `get_libname()` cannot resolve `libupsclient`, `nutscan_init()` falls back to `lt_dlopen()` by name. Two of those attempts stored their result in `nutscan_avail_xml_http` instead of `nutscan_avail_nut`, so the XML/HTTP scan availability ended up following the `libupsclient` load rather than the `libneon` one. Also guard the unversioned attempt like its neighbours, and fix two `#endif` comments naming the wrong macro.

2.61 2026-09-16 Jim Klimov <jimklimov+nut@gmail.com>

- `data/driver.list.in`: add entry for Microtek GI3KL 3000VA Online UPS [#3655]
- `NEWS.adoc`, `docs/nut.dict`, `drivers/nutdrv_qx_voltronic.c`: `NEWS.adoc`, `drivers/nutdrv_qx_voltronic.c`: add a note about, and bump version due to, added support for PI71 dialect [#3655]
- `tools/nut-scanner/nutscan-init.c`: skip work in vain on systems where `SONAME_LIB<X>` is same as `SOFILIB<X>` [#3654]

2.62 2026-09-15 user01010111 <lapses.50.booster@icloud.com>

- clients/upscmd.c, clients/upslog.c, clients/upsw.c, common/common.c, docs/developers.txt, docs/nut.dict, drivers/dstate.c, drivers/libshut.c, drivers/powercom-hid.c, drivers/richcomm_usb.c, drivers/snmp-ups.c, include/Makefile.am, include/nut-pragmas-type-limits-end.h, include/nut-pragmas-type-limits.h, server/upsd.c, tools/nut-scanner/scan_snmp.c: include: share type-limits diagnostic wrappers Move 17 repeated GCC diagnostic pairs into private repeat-includable headers. Preserve the push/pop condition, independent warning guards, and local Clang diagnostics in libshut.c. Register both headers for distribution and document their use. Refs: <https://github.com/networkupstools/nut/issues/3595> Prepared with OpenAI Codex using gpt-6-astra, xhigh reasoning.
- common/common-nut_version.c, common/common.c, docs/developers.txt, drivers/bcmxcp_usb.c, drivers/bestfcom.c, drivers/bestuferrups.c, drivers/genericups.c, drivers/nut-libfreeipmi.c, include/Makefile.am, include/nut-pragmas-unreachable-code-end.h, include/nut-pragmas-unreachable-code.h, server/netget.c, server/netmisc.c, server/netssl.c, tools/nut-scanner/nut-scanner.c, tools/nut-scanner/nutscan-init.c, tools/nut-scanner/scan_ipmi.c, tools/nut-scanner/scan_nut.c, tools/nut-scanner/scan_snmp.c, tools/nut-scanner/scan_xml_http.c: Extract shared unreachable-code diagnostic pragmas Move 17 identical GCC/Clang pragma pairs into private repeat-includable headers. Preserve the aggregate capability guard, diagnostic order and original boundaries; distribute the headers and document their use. Refs: <https://github.com/networkupstools/nut/issues/3595> AI assistance: OpenAI Codex gpt-6-astra (xhigh reasoning).
- clients/upsimage.c, clients/upsmon.c, clients/upssched.c, clients/upsset.c, common/common.c, docs/developers.txt, docs/nut.dict, drivers/belkinunv.c, drivers/bestfortress.c, drivers/dstate.c, drivers/mge-utalk.c, drivers/nutdrv_atcl_usb.c, drivers/richcomm_usb.c, drivers/serial.c, include/Makefile.am, include/nut-pragmas-format-nonliteral-end.h, include/nut-pragmas-format-nonliteral.h, tests/generic_gpio_ustest.c: Consolidate nonliteral format diagnostic pragmas Move 40 repeated GCC format-warning blocks into a private, repeat-includable header pair. Preserve each capability guard, diagnostic lifetime and the local Clang nesting in vupslog(). Distribute the headers and document their scope without changing calls or format-string handling. Refs: <https://github.com/networkupstools/nut/issues/3595> AI assistance: OpenAI Codex, gpt-6-astra (xhigh reasoning).
- clients/upsclient.c, docs/developers.txt, drivers/libhid.c, drivers/libshut.c, drivers/libusb0.c, drivers/libusb1.c, include/Makefile.am, include/nut-pragmas-range-checks-end.h, include/nut-pragmas-range-checks.h: Reuse paired pragma headers for full numeric range checks Move eight matching diagnostic blocks into repeat-includable private headers. Preserve the GCC capability guards, Clang branch, pragma order and all enclosed code. Register both headers for distribution and describe this warning family in the developer guide. Refs: <https://github.com/networkupstools/nut/issues/3595> Prepared with OpenAI Codex using gpt-6-astra at xhigh reasoning.

2.63 2026-09-15 user01010111 <12504630+user01010111@users.noreply.github.com>

- NEWS.adoc, docs/man/dummy-ups.txt, drivers/dummy-ups.c, tests/NIT/Makefile.am, tests/NIT/README.adoc, tests/NIT/repeater.py: dummy-ups: mark failed repeater refreshes stale Mark failed runtime refreshes stale, propagate variable-list termination errors, and reject short rows before accessing missing fields. Restore freshness after a complete response while preserving persistent retries and the strict/relaxed startup contract. Add optional loopback integration coverage through dummy-ups, upsd and upsc, and update the driver version, release notes and documentation. AI assistance: OpenAI Codex using gpt-6-astra.

2.64 2026-09-15 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, drivers/apc_common.c, drivers/apc_modbus.c: fix(apc_common): add missing USE_ON_DELAY to shutdown outlet commands apc_build_outlet_command()'s shutdown-family cases (SHUTDOWN_RETURN, SHUTDOWN_STAYOFF, SHUTDOWN_REBOOT_GRACEFUL) only set USE_OFF_DELAY, not USE_ON_DELAY. getgray (issue #3587) traced PowerChute's own native shutdown call chain on a Smart-UPS X 1500 down to the actual USB write and captured the real OutletCommand_BF values it sends: 0x1EC8 = OUTPUT_SHUTDOWN | USE_ON_DELAY | USE_OFF_DELAY | targets | source 0x1EC4 = OUTPUT_OFF | USE_ON_DELAY | USE_OFF_DELAY | targets | source 0x1ED0 = OUTPUT_REBOOT | USE_ON_DELAY | USE_OFF_DELAY | targets | source Without USE_ON_DELAY the shutdown itself still happens, but the switched outlet groups do not sequence back on at AC restore - this was the original symptom reported in #3587 ("no outlet group (OG) on sequencing"). All three computed values now match getgray's captures bit for bit when targeting every switched group. This function is shared with apc_modbus.c (both call it to build the same conceptual OutletCommand_BF/register value), so the fix and the apc_modbus DRIVER_VERSION bump travel together even though the actual hardware evidence

here is MicroLink/USB-transport only; apc_modbus's behavior on this path has not been independently verified. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com>

- NEWS.adoc, drivers/apcmicrolink.c: fix(apcmicrolink): read-modify-write the beeper alarm-delay register beeper.enable/beeper.disable wrote a bare 1 or 2 straight to 2:4.B.3A, overwriting the whole register. getgray tested this on a real SMX1500RM2U (issue #3587) and found it accepted but had no effect: ups.beeper.status stayed unchanged. The reason: 2:4.B.3A packs an alarm-delay value alongside the enabled/disabled flag (matches the alarm-delay live value documented elsewhere for this usage, e.g. 0x00C1 on an SCL500). The flag has to be changed in place - read the register first, then OR in ENABLED(1)/DISABLED(2) and clear the other of the pair: enable: (current | 1) & ~2 disable: (current | 2) & ~1 every other bit - the delay - is left untouched. Writing a bare 1 or 2 zeroes those bits, which real hardware silently treats as an invalid write rather than applying it. This driver now reads the current value via the existing microlink_get_descriptor_map_bits() helper and does the same read-modify-write before sending it back. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com>

2.65 2026-09-15 Jordan Rodgers <com6056@gmail.com>

- NEWS.adoc, UPGRADING.adoc, docs/nut.dict, tools/nut-scanner/nutscan-init.c: nut-scanner: try the SONAME of a library before its versioned file name nut-scanner only knew the file names it saw at build time, such as libusb-1.0.so.0.6.0 and the -dev link libusb-1.0.so, plus a few hardcoded variants. The SONAME libusb-1.0.so.0, the one name that stays valid across ABI-compatible updates and the only versioned name a runtime-only package installs, was not in the list. When Fedora 43 and Alpine 3.24 moved libusb from 1.0.29 to 1.0.30 (file renamed from .so.0.5.0 to .so.0.6.0) nut-scanner reported "Cannot load USB library (libusb-1.0.so.0.5.0)" until the NUT package was rebuilt, while the drivers, which link libusb-1.0.so.0 normally, kept working. Try SONAME_LIB* first in both the get_libname() pass and the lt_dlopen() fallback pass for every library that provides one (libusb 0.1 and 1.0, Net-SNMP, neon, Avahi, GLib/GIO, FreeIPMI). neon and Net-SNMP had the same latent problem (libneon.so.27.9.0, libnetsnmp.so.45.0.0). Also adds the words this introduces into docs prose to docs/nut.dict. See issue #3474.
- m4/ax_realpath_lib.m4, m4/nut_check_libavahi.m4, m4/nut_check_libfreeipmi.m4, m4/nut_check_libglib.m4, m4/nut_check_libneon.m4, m4/nut_check_libnetsnmp.m4, m4/nut_check_libusb.m4: m4: add AX_SONAME_LIB and record SONAME_LIB* for dynamically loaded libraries AX_REALPATH_LIB resolves a library to the fully versioned file present on the build system, e.g. /usr/lib/libusb-1.0.so.0.6.0, and that is the name nut-scanner remembers as SOPATH_LIB* and SOFILE_LIB* to dlopen() at run time. That file name can change on any upstream release of the library, including one that keeps the ABI and with it the SONAME (libusb-1.0.so.0). Add AX_SONAME_LIB which reads the SONAME recorded in the shared object (readelf or eu-readelf, objdump, elfdump on Solaris/illumos, otool -D on macOS, nothing on mingw where a DLL is loaded by its own file name) and call it from every nut_check_lib*.m4 that already uses AX_REALPATH_LIB, defining SONAME_LIB* next to the existing SOPATH_LIB* and SOFILE_LIB* values. A missing tool leaves SONAME_LIB* undefined.

2.66 2026-09-15 Jim Klimov <jimklimov+nut@gmail.com>

- tests/test_upsdconf.c: generate config once per check_value() [#3631]
- tests/test_upsdconf.c: range-limit "CERTREQUEST" [#3329, #3631]

2.67 2026-09-14 Jim Klimov <jimklimov+nut@gmail.com>

- server/conf.c, server/netssl.h: server/conf.c: range-check "CERTREQUEST" numeric values [#3631, #3640]
- common/state.c: clarify recently changed comment [#3630]
- NEWS.adoc: relocate entry for legrand-hid beeper fixes [#3639]
- UPGRADING.adoc: update about packaging NUT-Monitor [#3596]

2.68 2026-09-13 user01010111 <12504630+user01010111@users.noreply.github.com>

- UPGRADING.adoc, common/parseconf.c, docs/config-notes.txt, docs/man/dummy-ups.txt, tests/NIT/nit.sh, tests/NIT/parseconf-test.py: NIT: clarify parser watchdog failures and address review feedback Report query starts, watchdog expiry and captured process output. Allow a bounded per-query NIT_PARSECONF_TIMEOUT, defaulting to 60 seconds, and preserve per-client debug settings in the named parser testcase. Retain the parser error helper under #if 0. Clarify dummy-ups token joining, escaped spaces, Python compatibility and the quoted-hash upgrade behaviour. Keep EOF, multiline and active parser semantics unchanged. AI assistance: OpenAI Codex with gpt-6-astra.

2.69 2026-09-13 Jim Klimov <jimklimov+nut@gmail.com>

- ci_build.sh: iterate NUT_SSL_CLIENT_VALIDATION_VARIANTS values **under** non-trivial NUT_SSL_VARIANT builds [#3329, #1711]
- clients/upsmon.c: better differentiate when we yell NOTIFY_SUSPEND_FINISHED (only after one started) vs. NOTIFY_SUSPEND [#3405]

2.70 2026-09-12 Jim Klimov <jimklimov+nut@gmail.com>

- ci_build.sh: introduce NUT_SSL_CLIENT_VALIDATION_VARIANTS testing in BUILD_TYPE=default-all-errors* [#3329, #1711]

2.71 2026-09-11 Jim Klimov <jimklimov+nut@gmail.com>

- UPGRADING.adoc, conf/upsd.conf.sample, configure.ac, docs/configure.txt, docs/man/upsd.conf.txt, docs/security.txt: configure.ac, docs/*: --with-ssl-client-validation is no longer NSS-only [#3329]
- Makefile.am: evolve the NIT.env symlinking recipe a bit

2.72 2026-09-10 user01010111 <12504630+user01010111@users.noreply.github.com>

- scripts/perl/test_nut_parser.pl: Perl UPS::Nut tests: preload optional TLS before socket watchdogs Load optional IO::Socket::SSL before creating the plaintext socket fixture. The constructor also attempts this load with USESSL disabled; a slow first load can otherwise consume the fixture watchdogs before protocol exchange. Keep the existing deadlines, all 236 assertions and cleanup behavior. A controlled 20-second import reproduces the published script's server timeout and broken pipe; the updated setup completes all 236 assertions. Normal regression runs pass on macOS Perl 5.34.1, Linux Perl 5.40.1 and 5.005_04, and Strawberry Perl 5.42.3 under Wine/QEMU. Linux build, make check (12 Automake tests), spelling/style and distcheck-light pass. The CI log confirms the fixture timeout but does not establish import latency as its cause. The separate unchanged Python integration timeout remains unresolved. Wine is not Windows-kernel or MSYS2 acceptance. Related: #3634 AI assistance: OpenAI Codex with gpt-6-astra (high reasoning). The human contributor reviewed and authorized submission and DCO sign-off, and remains responsible for the contribution.
- tests/NIT/nit.sh: NIT: check JSON availability before parser integration tests Probe the selected Python interpreter for json before preparing the parser fixtures. Report a skip in the default suite if the prerequisite is absent, while preserving failure for an explicitly requested parser test group. Keep the existing parser and JSON assertions unchanged. Jenkins exposed a Python 3.7 installation unable to import its standard json module. The worker still needs repair to restore this integration coverage. Validated the missing-module failure and skip paths, native macOS and Linux parser integration, Linux make check (13 tests), style/spelling checks, and default make distcheck-light. The missing module was simulated locally; the exact Jenkins worker and physical UPS hardware were not tested. Related: #3633 AI assistance: OpenAI Codex with gpt-6-astra. The human contributor remains responsible for reviewing and submitting the change.
- configure.ac: configure: require PyNUT prerequisites before enabling Python tests When multiple Python interpreters are configured, enable each interpreter's PyNUT tests only if its existing runtime prerequisites and unittest are present. This avoids running tests on an interpreter lacking socket while PyNUT is enabled for another healthy interpreter. Reuse the existing per-interpreter checks, including their handling of a shared default interpreter. Validated with 11 configuration cases across Python

2.7 and Python 3, a Linux build, make check, installed and packaged PyNUT tests, and make distcheck-light. AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.

2.73 2026-09-10 Jim Klimov <jimklimov+nut@gmail.com>

- docs/net-protocol.txt: clarify relationship to RFC 9271 [#3637, #3411, #3359]
- docs/net-protocol.txt: clarify line-oriented nature of the protocol [#767]
- drivers/Makefile.am: comment about usbhid-ups dependency on serial [#3569]
- drivers/legrand-hid.c: switch DRIVER_VERSION to X.YY scheme [#3639]
- scripts/upsdrvsvcctl/nut-driver-enumerator.sh.in: fix the new \$(...) into old backticks [#3635, #3196] This script is expressly not necessarily parsed by BASH or shells with the newer POSIX syntax, but may be handled by "system shells" adhering to an older standard on some operating systems. So while backticks are less flexible than parentheses, they are more portable. Some earlier revisions of this script missed the introduction of this regression, fixing now that these lines were touched in PR #3635.
- scripts/upsdrvsvcctl/nut-driver-enumerator.sh.in: refactor with upsconf_getValue_awk() to avoid questions about nested quoting [#3635, #3196]
- UPGRADING.adoc: NDE syntax parser fix: move the section lower, refer to PR [#3635]
- NEWS.adoc: NDE syntax parser fix: move the section lower, refer to PR [#3635]
- scripts/upsdrvsvcctl/nut-driver-enumerator.sh.in: fix nested double-quotes missed earlier [#3196, #3635]
- docs/man/nut-driver-enumerator.txt: fix markup (line lengths/breaks) [#3635]

2.74 2026-09-10 Anton Popov <anton.a.popov@gmail.com>

- docs/nut.dict: docs: accept lowercase Legrand in spellcheck Aspell reports the brand name from NEWS.adoc as lowercase during spellcheck. Add the lowercase form alongside the existing proper-noun entry and keep the dictionary count synchronized.
- docs/nut.dict: docs: count Legrand in spelling dictionary The Legrand entry is already present in alphabetical order, but the personal dictionary header was one word short. Update the declared count to keep make spellcheck and distribution checks consistent.
- NEWS.adoc, drivers/legrand-hid.c: legrand-hid: fix beeper.disable to disable, not mute AudibleAlarmControl value 3 means "muted" (silences only the current alarm) per NUT's beeper_info table; value 1 means "disabled". The subdriver used 3 for beeper.disable, so the buzzer re-armed after every alarm and the command had no lasting effect. Map beeper.disable to 1, add beeper.mute for 3, and add the ups.beeper.status reading so the state is observable. Tested in Docker with a clean Ubuntu 24.04 build of usbhid-ups. Hardware validation was performed on a Legrand KEOR PDU 800 (1cb0:0038). Fixes: #3638

2.75 2026-09-09 user01010111 <12504630+user01010111@users.noreply.github.com>

- scripts/upsdrvsvcctl/nut-driver-enumerator.sh.in: nut-driver-enumerator: recognize whitespace with old mawk Use the explicit six C whitespace bytes in the value lexer because Trusty's mawk 1.3.3 does not implement the named whitespace class. Preserve the existing C locale, token rules and test expectations. The existing Trusty runtime tests improve from 133 pass/3 fail to 136/0 under each of sh, bash and dash. A 256-byte oracle, modern shell/awk checks, build, make check, staged package checks, spelling/style and complete-manual distcheck-light pass. These runtime checks do not reconstruct the exact Jenkins image; authenticated Jenkins failure logs remain unavailable, so this does not claim every CI error is resolved. Related: #3635 AI assistance: OpenAI Codex with gpt-6-astra (high reasoning). The human contributor remains responsible for reviewing and submitting the change.

- `scripts/perl/test_nut_parser.pl`: Perl UPS::Nut tests: support Windows paths and pseudofork Use a portable INCLUDE fixture while checking quote and backslash decoding at recursive AuthConf dispatch. Flush TAP before fork, exit the child normally and keep temporary-directory cleanup owned by the parent so the socket fixture completes under Windows pseudofork. Preserve timeouts, collision refusal and production behavior. Validated with all 236 assertions on macOS Perl 5.34.1, Linux 5.40.1 and 5.005_04, and Strawberry 5.42.3 under Wine/QEMU. Build, make check, spelling/style and distcheck-light pass. Wine does not establish exact AppVeyor/MSYS2 or Windows-kernel acceptance; MSYS2 Perl could not start under Wine before loading NUT. Related: [#3634](#) AI assistance: OpenAI Codex with gpt-6-astra (high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, UPGRADING.adoc, docs/man/nut-driver-enumerator.txt, scripts/upsdrvsvcctl/nut-driver-enumerator.sh.in, tests/nut-driver-enumerator-test—ups.conf, tests/nut-driver-enumerator-test.sh: nut-driver-enumerator: preserve and decode ups.conf values once Preserve quotes and backslashes during ups.conf normalization, then decode the first value token when queried. Keep full and section caches consistent for values, media classification and checksums, including quoted multiline values. Document possible one-time reconciliation after upgrading. Extend existing self-tests and fixtures. Validated across macOS/Linux shells and awk implementations, corrected-C comparisons, generated/staged packages, documentation checks and distcheck-light. Related: [#3629](#) AI assistance: OpenAI Codex with gpt-6-astra (high reasoning) and gpt-daybreak-blue-latest (high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, UPGRADING.adoc, scripts/Makefile.am, scripts/perl/UPS/Nut.pm.in, scripts/perl/test_nut_parser.pl: Perl UPS::Nut: decode NUT tokens and quote authentication arguments Decode NUT reply and configuration tokens once, including escaped INCLUDE paths, and quote literal authentication credentials. Preserve public return shapes and accept the actual CMDDESC response. Update POD, upgrade notes and the module version. Add a core-module regression script to check/distribution rules. Validated on Perl 5.005_04, 5.34.1 and 5.40.1, corrected-C comparisons, localhost upsd/dummy-ups, generated/installed modules and distcheck-light. Related: [#3629](#) AI assistance: OpenAI Codex with gpt-6-astra (high reasoning) and gpt-daybreak-blue-latest (high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, common/parseconf.c, docs/config-notes.txt, docs/man/dummy-ups.txt, docs/net-protocol.txt, tests/Makefile.am, tests/NIT/Makefile.am, tests/NIT/nit.sh, tests/NIT/parseconf-test.py, tests/parseconf-test.c: parseconf: accept literal hashes inside double quotes Accept quoted hashes in the shared parser while retaining unquoted comments and escaped output for older peers. Keep default upsc text output unchanged and document configuration, protocol and dummy-ups behavior. Add portable parser regressions and localhost text/JSON integration. Validated with C89/C99/C11 GCC and Clang builds, sanitizers, full NIT, package/install checks, documentation checks and distcheck-light. Related: [#607](#), [#1306](#), [#3629](#) AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning) and gpt-daybreak-blue-latest (high reasoning). The human contributor remains responsible for reviewing and submitting the change.

2.76 2026-09-09 Jim Klimov <jimklimov+nut@gmail.com>

- README.adoc: fix back the graph to star-history.com See <https://github.com/star-history/star-history/issues/539>, <https://github.com/orgs/community/discussions/201209> outcry, and resulting <https://github.blog/changelog/2026-09-04-new-api-endpoint-provides-privacy-safe-star-history-data/>
- `scripts/python/module/PyNUT.py.in`: fix reconnection attempts after SSL handshake timeout [[#3329](#)] The original implementation did try, but the socket was closed so the first retry quickly failed with "Bad file descriptor".
- `scripts/python/module/test_nutclient.py.in`: work around lack of `unittest` module on some systems [[#3627](#)]
- `tests/pynut-commands-test.py`: add a top comment about lack of shebang [[#3628](#)]
- `tests/Makefile.am`: refactor `pynut-commands-test*` goals to consider availability of `unittest` module [[#3628](#)]
- `docs/nut.dict`, `scripts/python/README.adoc`: `scripts/python/README.adoc`: refer to `tests/pynut-commands-test.py` and `unittest` module [[#3628](#)]
- `configure.ac`: detect ability to `unittest` current Python ecosystem [[#3627](#), [#3628](#)]

2.77 2026-09-08 user01010111 <12504630+user01010111@users.noreply.github.com>

- tests/test_upsdconf.c: tests: make upsd tracking stub safe to format Return a static printable value after recording an unexpected tracking_get call. A NULL result lets GCC at -O2 diagnose the included sstate_readline format argument and fail builds that treat warnings as errors. The failure counter still records any unexpected use of the linkage stub. AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, UPGRADING.adoc, docs/man/upsd.conf.txt, server/conf.c, server/netssl.c, server/sstate.c, tests/.gitignore, tests/Makefile.am, tests/test_upsdconf.c: upsd: validate numeric configuration values Use the existing str_to_int/str_to_long helpers for MAXAGE, TRACKINGDELAY, MAXCONN and CERTREQUEST. Check MAXCONN for narrowing to nfds_t, and commit converted values only after successful validation. Preserve public types, supported numeric spelling, named certificate levels, and existing configuration error handling. Avoid signed overflow when applying MAXAGE's grace second at INT_MAX. Restore a missing brace in the NSS certificate range check so the affected client-validation build can be exercised. Add configuration loading and boundary regression coverage to make check, and document the malformed-value compatibility change under upsd. Closes: #676 AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- tests/statetest.c: tests: check state node before dereferencing Stop statetest with a clear failure message if the required device.contact lookup returns NULL, before reading the node name or changing its flags. AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, common/state.c, tests/Makefile.am, tests/NIT/nit.sh, tests/statetest.c: common: preserve case-only state value updates Compare raw state values byte-for-byte so mutable updates such as device.contact changing from Operations to operations are retained and published. Preserve case-insensitive variable-name matching, immutable values, timestamps and metadata. Add common state-library regression coverage and an existing NIT dummy sequence check for both case transitions. Document the correction in NEWS. Closes: #3622 AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.

2.78 2026-09-08 Jim Klimov <jimklimov+nut@gmail.com>

- Makefile.am: actually generate a (top_dir)/NIT.env symlink to help with make check-NIT* runs [#1711]
- .gitignore: GitIgnore a (top_builddir)/NIT.env file that can be symlinked to simplify dev-testing ... and other repetitive build/-configuration aids

2.79 2026-09-07 user01010111 <12504630+user01010111@users.noreply.github.com>

- NEWS.adoc, scripts/python/module/Makefile.am, scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in: PyNUT: restore socket cleanup in disconnect() Remove the unmangled hasattr() check which prevented cleanup of a retained client's socket. The class-level None default already makes direct access safe before a connection has been assigned. Preserve the existing best-effort LOGOUT, close and reference cleanup behavior. Exercise it in the existing Python test script and make check, including failure injection, partial initialization and a loopback socket. Document the fix in NEWS.adoc. Closes: #3619 AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change.
- NEWS.adoc, scripts/python/module/PyNUT.py.in, tests/Makefile.am, tests/pynut-commands-test.py: PyNUT: use command identifiers in description requests Decode LIST CMD identifiers as ASCII for GET CMDDESC requests and response offsets, preserving the public bytes dictionary and description fallback. Add request-sensitive regression coverage to make check and note the fix in NEWS.adoc. Validated with Python 2.6, 2.7 and 3.4 through 3.14, localhost upsd with dummy-ups, generated/installed/packaged modules, make check, spellcheck and distcheck-light. AI assistance: OpenAI Codex with gpt-6-astra (x-high reasoning). The human contributor remains responsible for reviewing and submitting the change. Closes: #3621

2.80 2026-09-07 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/app/locale/NUT-Monitor.pot, .../python/app/locale/fr/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/fr/fr.po, .../python/app/locale/it/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/it/it.po, .../python/app/locale/ru/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/ru/ru.po, scripts/python/app/ui/aboutdialog1.ui, scripts/python/app/ui/gui-1.3.glade, scripts/python/app/ui/gui-1.3.glade.h: scripts/python/app/*: ACTUALLY bump the version to 2.2.0 as seen in UI resources [#3596, #3616, #3625]

- scripts/python/app/NUT-Monitor-py2gtk2.in: bump version to 2.2.0 [#3616, #3625]
- scripts/python/.gitignore: GitIgnore outputs of the app-specific test suite in scripts/python [#3596]
- configure.ac: comment about one NUT-Monitor-py3qt.in producing two script files [#3596]
- scripts/python/app/NUT-Monitor-py3qt.in: note other achievements in version 2.2.0 [#3616, #3625]
- NEWS.adoc: move note about PyNUTClient fix to client libraries section [#3627]
- scripts/python/app/NUT-Monitor-py3qt.in: move note about dual-generation to top of file [#3596]
- NEWS.adoc: move PyQt5 and PyQt6 NUT-Monitor merger note to the client section [#3596]
- NEWS.adoc: document NUT-Monitor fixes [PRs #3516, #3525]
- tests/nutclienttest.cpp: avoid commented-away debug tracing code, use NUT_DEBUG_LEVEL instead [#3439]
- docs/nut.dict: update to cover preceding PR merge train
- clients/upsimage.c, clients/upsset.c, clients/upsstats.c: clients/{upsset,upsstats,upsimage}.c: return the debug ability

2.81 2026-09-06 nmbro <nmbro@users.noreply.github.com>

- NEWS.adoc, docs/nut.dict: NEWS.adoc: reword to avoid a dictionary addition Prefer rewording over growing docs/nut.dict for a one-off word. Drop "decompiling" (revert the dict entry from the previous commit) and reword around it; also avoid "PowerChute's", which aspell rejected as a possessive of the dictionary word "PowerChute" even though the plain word is already listed. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code-session_01LB4rN1EMA4s2TumED2ZWB2
- docs/nut.dict: add "decompiling" CI's clean aspell (no personal word list) flagged it in the new NEWS.adoc beeper.mute bullet: *decompiling PowerChute's* . . . Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code-session_01LB4rN1EMA4s2TumED2ZWB2
- NEWS.adoc: document outlet-group-count and beeper fixes Cover the three commits added to the PR #3591 series after it opened: outlet.group.count no longer requiring an unswitched main group, ups.beeper.status decoded as a bitmask, and beeper.mute replaced by beeper.enable/beeper.disable. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code-session_01LB4rN1EMA4s2TumED2ZWB2
- docs/man/apcmicrolink.txt, drivers/apcmicrolink.c: fix(apcmicrolink): replace beeper.mute with beeper.enable/disable beeper.mute wrote APC_USER_IF_CMD_MUTE_ALL_ACTIVE_AUDIBLE_ALARMS to 2:4.B.3B, a "user interface command" register ported from apc_modbus by analogy when this driver was first written. It had no confirmed effect on real Microlink hardware - getgray reported no observable change on an SMX1500RM2U (issue #3587), and the previous commit's digging into ups.beeper.status turned up more evidence the apc_modbus-style command register isn't what real client software uses here. Decompiling PowerChute's own CompositeAudibleAlarm class settles it: its alarm on/off control (ChangeAudibleAlarmRunnable) writes a plain value - 1 for enabled, 2 for disabled - straight to the AlarmDelay usage (2:4.B.3A), the exact same usage ups.beeper.status reads. That is a persistent setting, not the transient "silence the currently sounding alarm, then re-arm automatically" action NUT documents beeper.mute as (data/cmdvartab: "Temporarily mute the UPS beeper"). Rather than keep a command named "mute" that actually disables the beeper until told otherwise, drop it and expose what the device genuinely supports: beeper.enable (writes 1) and beeper.disable (writes 2), both to 2:4.B.3A. test.panel.start is untouched - getgray already confirmed that command works as-is, still via the 2:4.B.3B "user interface command" register. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code-session_01LB4rN1EMA4s2TumED2ZWB2
- drivers/apcmicrolink-maps.c: fix(apcmicrolink): read ups.beeper.status as a bitmask, not an enum ups.beeper.status matched the exact values 193/194 (0xC1/0xC2), taken from round-tripping PowerChute writes on one SCL500RM1UC. Decompiling PowerChute's own CompositeAudibleAlarm class shows this is actually a bitmask: bit 0 (0x01) means enabled, bit 1 (0x02) means disabled. The 0xC1/0xC2 values only worked because that particular unit always sets two extra bits (0xC0) alongside the enabled/disabled bit. A second device (APC Smart-UPS X1500, SMX1500RM2U, reported in <https://github.com/networkupstools/nut/issues/3587>) reports 0x09

- bit 0 (enabled) plus bit 3, PowerChute's unrelated "advanced menu" UI flag - which never matched either exact value, so ups.beeper.status published the raw number instead of a label. Switch the map to MLINK_DESC_BITFIELD_MAP (already used elsewhere in this driver for the same kind of field) with the real bit values. Verified against real captures from both devices: still "enabled" on the SCL500RM1UC, now "enabled" instead of "9" on the SMX1500RM2U. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code/session_01LB4rN1EMA4s2TumED2ZWB2
- docs/man/apcmicrolink.txt, drivers/apcmicrolink.c: fix(apcmicrolink): count switched outlet groups without a main group microlink_outlet_group_count() returned 0 whenever the unswitched "main" group (descriptor path 2:4.3E.B6) was absent, even if switched groups (2:4.3D[i].B6) were present. This silently skipped outlet.group.count, outlet.group.N.switchable, and all outlet instant commands on any device without an unswitched group, while the per-group name/status/delay data kept publishing fine through the separate descriptor-export path - a real device (APC Smart-UPS X1500, SMX1500RM2U) confirmed this: three switched groups with no group 0. Split the group-presence check per index (microlink_outlet_group_exists()) instead of gating everything on group 0, and use it everywhere group membership is tested: command registration, outlet.group.N validation, and the "all groups" target mask for load./shutdown. commands. Devices that do have a group 0 are unaffected (verified against a real SCL500RM1UC capture, byte-for-byte identical output before and after). Use of coding helper tools and AI disclosed: Claude Code was used for development assistance. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code/session_01LB4rN1EMA4s2TumED2ZWB2

2.82 2026-09-06 Laurent Bigonville <bigon@bigon.be>

- scripts/python/app/NUT-Monitor-py3qt6.in: nut-monitor: py3qt6: Fix unscoped QStandardPaths enums for PyQt6 PyQt6 requires fully-scoped enum access, unlike PyQt5 which allowed the shorthand QStandardPaths.AppDataLocation form. The unscoped references were missed during the Qt6 port and would raise AttributeError, breaking resource lookup (ui/pixmaps/icons) whenever the app runs from an installed location rather than the source tree. Fixes: [#3624](#)
- scripts/python/app/NUT-Monitor-py3qt6.in: nut-monitor: py3qt6: Fix DesktopFileName to match the Qt6 desktop file Was copy-pasted from the Qt5 variant, causing the Qt6 build to misidentify itself to the window manager/taskbar as nut-monitor-py3qt5 instead of nut-monitor-py3qt6.

2.83 2026-09-06 Jim Klimov <jimklimov+nut@gmail.com>

- docs/nut-names.txt: document that input.sensitivity is an opaque string [[#3591](#)] Per investigation of current code posted in <https://github.com/networkupstools/nut/pull/3591#issuecomment-5558864820> common values include: auto, low, reduced, medium, normal, high in various order and with 2 to 4 of these present in particular drivers; also H, M, L; also Normal, Reduced, Low (cased).

2.84 2026-09-05 user01010111 <lapses.50.booster@icloud.com>

- drivers/snmp-ups.c, tests/Makefile.am, tests/snmp-ups-setvar-test.c: tests: strengthen snmp-ups setvar regression Assert the exact OID, Net-SNMP type, value and call count passed by su_setOID() for successful and failed setters. Link the focused test against only snmp-ups.c and snmp-ups-helpers.c by supplying nut_snmp_set() at its existing public boundary. Production builds retain the real setter and complete MIB registry. AI assistance: OpenAI Codex GPT-6 (controller) and gpt-5.6-sol at high reasoning (implementation, validation and review). The human contributor remains responsible for reviewing and validating the change.

2.85 2026-09-04 Jonathan Chavez <jonathangiancarlo1@gmail.com>

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-*.in: fix AttributeError in gui_updater.attempt_reconnect() [[#3509](#), [#3530](#)] *The reconnection code added in [#3530](#) reads the "ups_authentication_check" widget through self.__widgets, but __attempt_reconnect() lives in the gui_updater class, which has no __widgets attribute (name-mangled to _gui_updater__widgets). The widgets belong to the interface class and are reached, as the surrounding lines already do, through self.__parent_class*

The branch is taken whenever `PyNUT.AuthConf.getAuthConf()` returns an object, which is the common case, so every reconnection attempt raised: `AttributeError: gui_updater object has no attribute _gui_updaterwidgets` In the Qt variants an exception escaping a slot is fatal: PyQt turns it into `qFatal()` and the process aborts (SIGABRT) instead of reconnecting. Reproduced on Arch Linux, PyQt6 6.11 / Qt 6.11.2, by restarting `upsd` while NUT-Monitor-py3qt6 (master, 59fad7a97) was connected. Same one-line fix applied to the py3qt5, py3qt6 and py2gtk2 variants. Co-Authored-By: Claude Fable 5.1 <noreply@anthropic.com>

2.86 2026-09-04 Jiří Setnička <setnicka@seznam.cz>

- NEWS.adoc, drivers/mge-hid.c: mge-hid: add Eaton 5P Gen2 models to `mge_model_names` The Gen2 units report iModel strings like "650iR G2", which match no entry in `mge_model_names[]`, so `mge_type` stays `MGE_DEFAULT`. Both `mge_battery_voltage` and `mge_battery_voltage_nominal_fun()` return NULL for unknown types, so `battery.voltage` and `battery.voltage.nominal` are dropped although their `UPS.PowerSummary.Voltage` and `.ConfigVoltage` source paths are parsed fine. Names are assumed per the VA numbers and form factors of the Gen2 range, as was done before for the 5PX and 5SC series. Bump the subdriver version so `driver.version.data` reflects the extended mapping table.

2.87 2026-09-02 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, UPGRADING.adoc, clients/upsmon.c, clients/upsstats.c, conf/hosts.conf.sample, conf/upsmon.conf.sample.in, docs/man/hosts.conf.txt, docs/man/upsmon.conf.txt: clients/upsmon.c, client/upsstats.c, conf/, docs/man/: support special values of `AUTHCONF` option like command-line clients do [#3439, #3329]
- clients/authconf.c, conf/hosts.conf.sample, conf/upsmon.conf.sample.in, docs/man/hosts.conf.txt, docs/man/nutauth.conf.txt, docs/man/upsc.txt, docs/man/upsccli_read_authconf_file.txt, docs/man/upscmd.txt, docs/man/upslog.txt, docs/man/upsmon.conf.txt, docs/man/upswr.txt, tests/test_authconf.c: clients/authconf.c, conf/, docs/man/: `parse_authconf_file()`: treat non-absolute paths as relative to `confpath()` [#3439, #3329]

2.88 2026-09-01 user01010111 <lapses.50.booster@icloud.com>

- common/snprintf.c: keep magnitudes unsigned Carry unsigned format arguments in unsigned LONGLONG from `dopr()` through `fmtint()`, rather than converting them through a possibly narrower signed long. Track negativity separately for signed decimal values and keep the magnitude negation in unsigned arithmetic so `LONGLONG_MIN` remains defined. Convert pointers through `size_t` before widening, and extend the built-in comparisons with narrow and wide unsigned extrema. AI assistance: OpenAI Codex gpt-5.6-sol with xhigh reasoning was used for repository analysis, implementation, review, drafting and validation. Anthropic Claude Fable 5 (claude-fable-5) with xhigh reasoning performed an independent adversarial review and produced a prototype used as design evidence. The human contributor reviewed the change and remains responsible for it.
- common/snprintf.c: explain unsigned negation Document why `fmtint` negates the unsigned magnitude when formatting negative values, including `LONGLONG_MIN`. AI assistance: OpenAI Codex gpt-5.6-sol was used for repository analysis, implementation, review, drafting and validation. The human contributor reviewed the change and remains responsible for it.
- NEWS.adoc, drivers/Makefile.am, tests/Makefile.am, {drivers ⇒ tests}/snmp-ups-setvar-test.c: tests: relocate snmp-ups setvar regression Move the focused SNMP setvar regression from `drivers/` to `tests/` and wire it through the existing linked-driver-source pattern. Restore `drivers/Makefile.am` to the normal driver-only build and add the requested NEWS entry for issue #3360. AI assistance: OpenAI Codex gpt-5.6-sol at high reasoning. The human contributor remains responsible for reviewing and validating the change.
- NEWS.adoc, tests/Makefile.am, tests/become-user-root-warning-test.sh: tests: improve root-warning test portability Honour `EXEEXT`, pass the exact build-tree `upslog` path, and use the current `UID-0` account name. Treat an empty `TMPDIR` as unset and add the requested NEWS entry. Refs #3471. AI assistance: OpenAI Codex gpt-5.6-sol at high reasoning was used for investigation, implementation and validation. The human contributor remains responsible for the change.
- drivers/Makefile.am, drivers/snmp-ups-setvar-test.c, drivers/snmp-ups.c: drivers/snmp-ups: prefer exact outlet mappings Look up exact mappings before interpreting outlet names as numbered templates. Return the existing unknown status when a numbered template is absent, update server-side writable values through `dstate`, and track ownership for instantiated entries. Add a focused hardware-free regression for exact mappings, missing templates, valid numbered templates, unavailable mappings

and cleanup. AI assistance: OpenAI Codex gpt-5.6-sol at high reasoning. The human contributor remains responsible for reviewing and validating the change.

- common/common.c, tests/Makefile.am, tests/become-user-root-warning-test.sh: common: warn when become_user() leaves root credentials Check the real and effective UIDs after the POSIX credential transition, and warn once if either remains zero. Keep non-root starts and the intentionally privileged upsmon -p path unchanged. Fixes [#3471](#). AI assistance: OpenAI Codex gpt-5.6-sol at high reasoning was used for investigation, implementation and validation. The human contributor remains responsible for the change.
- AGENTS.md, docs/developers.txt: docs: wrap agent guidance source lines Wrap agent guidance source lines consistently with existing NUT documentation while preserving the rendered text. AI assistance: OpenAI Codex gpt-5.6-sol was used for drafting and validation. The human contributor remains responsible for the change.
- AGENTS.md, Makefile.am, docs/developers.txt: docs: add instructions for coding agents Add a root AGENTS.md which directs coding agents to existing NUT contribution requirements and recurring project-specific review checks. Tell contributors using coding agents to ensure their tool reads the file, and register it for source distribution. AI assistance: OpenAI Codex gpt-5.6-sol was used for research, drafting and review. The human contributor remains responsible for the change.
- NEWS.adoc, drivers/apc-mib.c: apc: expose NMC3 ambient temperature APC Network Management Card 3 units expose universal I/O temperature at uioSensorStatusTemperatureDegC, but the APC subdriver does not map that OID to ambient.temperature. Add the OID as the first ambient provider, reject the MIB's -1 invalid sentinel, bump APCC_MIB_VERSION to 1.62, and document the addition. Fixes [#3250](#)
- clients/upssched.c, common/wincompat.c, drivers/dstate.c, include/wincompat.h: win32: restrict named-pipe access Replace the shared present NULL DACL with an explicit ACL granting pipe access to the current process identity, LocalSystem and Administrators. Keep PIPE_REJECT_REMOTE_CLIENTS and apply the same descriptor to initial and replacement daemon, event, driver-state and upssched listeners. Include FILE_CREATE_PIPE_INSTANCE for replacement listener creation, and release the temporary ACL after CreateNamedPipe returns. AI assistance: OpenAI Codex gpt-5.6-sol with high reasoning coordinated and reconciled the work. OpenAI Codex gpt-daybreak-blue-latest with high reasoning was used for security analysis, implementation, adversarial review, cross-toolchain validation, native Windows validation and contribution drafting. The contributor reviewed the change and remains responsible for its quality and content.
- NEWS.adoc, scripts/python/Makefile.am, scripts/python/app/NUT-Monitor-py3qt.in: NUT-Monitor: address review follow-ups Name the shared Qt template variable consistently, document how configure generates the two legacy command names, and record version 2.2.0. Link the NEWS entry to [PR #3596](#). OpenAI Codex gpt-5.6-sol assisted with implementation and validation. The delegated worker used high reasoning. The human contributor reviewed and validated the change and remains responsible for it.

2.89 2026-09-01 Yifeng Li <tomli@tomli.me>

- UPGRADING.adoc: huawei-ups2000 instant commands and Modbus fixes [[#3593](#), [#3603](#)]
- NEWS.adoc: huawei-ups2000 instant commands and Modbus fixes [[#3593](#), [#3603](#)]

2.90 2026-09-01 Jim Klimov <jimklimov+nut@gmail.com>

- tests/nutclienttest.cpp: test_ssl_context_registry(): adapt to builds without OpenSSL backend [[#3439](#)]
- clients/nutclient.cpp: fix builds without OpenSSL [[#3439](#)]
- clients/upsmon.c: revise debug-logging of notifycmd_concat
- clients/upsclient.c: upscli_tryconnect(): debug-log getting into an SSL attempt [[#3439](#)]
- clients/upsclient.c, clients/upslog.c, clients/upsstats.c, docs/new-clients.txt: clients/upsclient.c, clients/upslog.c, clients/upsstats.c: upscli_sslinit(): search for suitable ups→ssl_ctx if missing [[#3439](#)] An older hypothesis about pre-assigning ups→ssl_ctx was wrong because upscli_tryconnect() wipes the ups contents as heap trash.
- NEWS.adoc: mention new AGENTS.md and SECURITY.md [[#3470](#), [#3608](#)]

2.91 2026-08-31 user01010111 <lapses.50.booster@icloud.com>

- `drivers/bestups.c`: `bestups`: validate fixed-format Q1 replies Validate the complete 46-byte Q1 layout before periodic polling or shutdown copies and interprets its fields. Use one driver-local parser for both entry points so malformed field boundaries, embedded NUL bytes and unexpected whitespace are rejected without partial state publication. Preserve legacy fixed-width content tolerance, shutdown retries and the unconditional shutdown command. Bump the driver version to 1.13. AI assistance: OpenAI Codex gpt-5.6-sol (high reasoning) assisted with implementation and validation. OpenAI gpt-daybreak-blue-latest (high reasoning) performed independent security-boundary and candidate reviews.
- `drivers/metasys.c`: `metasys`: validate UPS_INFO response length Reject UPS_INFO payloads outside the protocol-defined range before accessing response fields or copying the serial number. Derive the copy limit from the destination buffer and reserve space for the terminator. This preserves valid 1-12-byte serial numbers, leaves generic frame handling unchanged, and bumps the driver version to 0.15.
- `NEWS.adoc`, `drivers/xppc-mib.c`: `xppc`: map inverter standby to OL Line-interactive XPPC devices can report `standBy(8)` while utility power supplies the load and only the inverter is idle. Mapping that value to OFF makes `upsmon` treat a healthy UPS as not supplying power. Map `standBy(8)` to OL, clarify the inverter-versus-UPS distinction, bump the XPPC MIB version to 0.42, and document the correction. Fixes [#3512](#)
- `clients/nutclient.cpp`, `common/common.c`, `common/nutconf.cpp`, `common/nutstream.cpp`, `common/nutwriter.cpp`, `drivers/adel-system_cbi.c`, `drivers/apc_common.c`, `drivers/apc_modbus.c`, `drivers/apcmicrolink.c`, `drivers/dummy-ups.c`, `drivers/generic_modbus.c`, `drivers/liebert-gxe.c`, `drivers/main.c`, `drivers/netxml-ups.c`, `drivers/nutdrv_qx.c`, `drivers/phoenixcontact_modbus.c`, `drivers/up-scode2.c`, `drivers/usbhid-ups.c`, `include/Makefile.am`, `include/nut-pragmas-covered-switch-default-end.h`, `include/nut-pragmas-covered-switch-default.h`, `server/upsd.c`, `tools/nut-scanner/scan_avahi.c`, `tools/nutconf/nutconf-cli.cpp`: `include`: share covered-switch pragmas [#3595](#) Move the repeated GCC and Clang diagnostic push/pop sequence for exhaustive switches with defensive default clauses into intentionally repeat-includable paired headers. Replace only the 41 regions that use the same suppression family and distribute the headers as private build inputs. Leave the mixed `-Wswitch-enum` case and compiler-specific regions local. Refs [#3595](#) AI assistance: OpenAI Codex gpt-5.6-sol was used for repository analysis, implementation, review, drafting and validation with high reasoning. The human contributor remains responsible for the contribution.
- `common/snprintf.c`: preserve wide integer values Keep fallback integer formatting at LLONG width. This prevents pointers and long-long values from being truncated through long on LLP64 systems. Handle the signed minimum without overflow and extend `TEST_SNPRINTF` with pointer and long-long regression comparisons. Fixes [#1602](#) AI assistance: OpenAI Codex gpt-5.6-sol was used for repository analysis, implementation, review, drafting and validation. The human contributor reviewed the change and remains responsible for it.
- `clients/cgilib.c`, `clients/cgilib.h`, `clients/upsimage.c`, `clients/upsset.c`, `clients/upsstats.c`, `conf/upsstats-modern-single.html.sample.in`, `docs/man/upsstats.html.txt`, `docs/nut.dict`, `tests/Makefile.am`, `tests/test_cgilib.c`: `clients`: apply contextual encoding to CGI output Encode dynamic values for their HTML, URL component, JSON, or numeric CSS contexts. Keep CGI debug records on `stderr` and add focused regression coverage for the shared encoders.
- `NEWS.adoc`, `configure.ac`, `docs/configure.txt`, `docs/man/NUT-Monitor.txt`, `scripts/python/Makefile.am`, `scripts/python/README.adoc`, `scripts/python/app/NUT-Monitor`, ... {`NUT-Monitor-py3qt6.in` $\Rightarrow$ `NUT-Monitor-py3qt.in`}, `scripts/python/app/NUT-Monitor-py3qt5.in`, `scripts/python/app/NUT-Monitor-test.sh`, `scripts/python/app/README.adoc`: `NUT-Monitor`: share PyQt implementation [#3589](#) Generate the PyQt5 and PyQt6 commands from one Python 3 template while preserving their installed names and backend-specific behaviour. Let the dispatcher prefer PyQt6 by default and support `NUT_MONITOR_QT=5` or `NUT_MONITOR_QT=6` for explicit selection. Add focused dispatcher coverage and update the build recipes and documentation. Fix the existing authenticated automatic-reconnect callback to read its authentication widget through the parent interface. This prevents the unhandled timer callback exception that aborted both PyQt bindings. OpenAI Codex gpt-5.6-sol assisted with implementation and validation. The delegated worker used high reasoning. The human contributor reviewed and validated the change and remains responsible for it.

2.92 2026-08-31 Yifeng Li <tomli@tomli.me>

- `docs/man/huawei-ups2000.txt`: document bug [#3593](#) and bug [#3603](#). Two bugs [#3593](#), [#3603](#) can prevent users from sending any instant commands or changing any variables at all. Document these known bugs in the man page.
- `drivers/huawei-ups2000.c`: `huawei-ups2000`: bump driver version to v0.14.

- drivers/huawei-ups2000.c: huawei-ups2000: use modbus_write_register, not modbus_write_registers, fix [#3593](#). On some Huawei UPS2000 models, no instant commands or variable writes are possible, because modbus_write_registers() doesn't work correctly. Upon investigation, Huawei UPS2000 doesn't support multiple-register writes at all (Modbus command 0x10), the official datasheet only supports single-register writes (Modbus command 0x06). Multiple-register writes worked on some models, but this turned out to be undefined behavior, it doesn't work with all models. This commit switches modbus_write_registers() to modbus_write_register() to fix the problem.
- drivers/huawei-ups2000.c: huawei-ups2000: fix broken instant command invocations, close [#3603](#). A serious regression was introduced during the "fightwarn" campaign that completely broke bypass.start, shutdown.return, shutdown.reboot, and shutdown.reboot.graceful. Any attempt to use them will fail, with an error logged to syslog: huawei-ups2000: instcmd: command [bypass.start] reg1 is negative huawei-ups2000: instcmd: command [shutdown.return] reg1 is negative huawei-ups2000: instcmd: command [shutdown.reboot] reg1 is negative huawei-ups2000: instcmd: command [shutdown.reboot.graceful] reg1 is negative Because the regression was introduced shortly after the initial driver was merged into the upstream, these instant commands do no work in any NUT version (with the exception of the earliest development build). This problem was not discovered during pre-merge test for the same reason: it's a regression that occurred after the Pull Request was merged. In huawei-ups2000, all instant commands are driven by a lookup table to consolidate the logic to a central dispatcher. Some commands are dispatched to a (*handler_func)(uint16_t reg1) function, which is used to handle commands that needs additional processing. Not all *handler_func actually reads "reg1", it's an optional argument. Because declaring the function to accept a variadic or a (void *) would be over-engineering, a simple unsigned 16-bit value is used, *handler_func may or may not use it. During the project-wide "fightwarn" refactor campaign, it was noticed that "reg1" is stored in the lookup table as "int16_t" (-1 indicates an unneeded register), but *handler_func accepts a "uint16_t". In case that *handler_func doesn't need the register ID from the lookup table, "reg1" is implicitly converted from -1 to 65535. This is harmless, since "reg1" is not actually used in this case. Unfortunately, in an attempt to make the code "safe" by suppressing compiler warning, an incorrect range check was added to the function, which refuses to invoke *handler_func if "reg1" is negative. As a result, all instant commands that don't use "reg1" are broken. These commands include bypass.start, shutdown.return, shutdown.reboot, and shutdown.reboot.graceful. To fix this problem, fix the implicit type conversion correctly by change the "reg1" data type in the lookup table from "int16_t" to "uint16_t", and remove the "negative reg1" check. As defensive programming. all *handler_func that actually need "reg1" would check and abort if "reg1" is 0. Fixes: e9f02e242acf ("drivers/huawei-ups2000.c: instcmd(): range-check and cast for ups2000_write_register() and handler_func()")

2.93 2026-08-31 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, UPGRADING.adoc, clients/upsmon.c, clients/upsstats.c, conf/hosts.conf.sample, conf/nutauth.conf.sample.in, conf/upsmon.conf.sample.in, docs/man/hosts.conf.txt, docs/man/nutauth.conf.txt, docs/man/upsmon.conf.txt: clients/upsmon.c, clients/upslog.c, NEWS.adoc, docs, conf/* samples: introduce AUTHCONF keyword [\[#3439\]](#)
- docs/man/Makefile.am, docs/man/upscli_get_or_create_ssl_context.txt, docs/man/upscli_init.txt, docs/man/upscli_set_ssl_context.txt, docs/man/upscli_ssl.txt, docs/man/upsclient.txt, docs/man/upslog.txt, docs/man/upsmon.txt, docs/man/upsstats.cgi.txt, docs/new-clients.txt, docs/nut.dict: docs/new-clients.txt, docs/man/ups*.txt: document multiple SSL context support [\[#3439\]](#)
- NEWS.adoc: document ability for multiple SSL contexts [\[#3439\]](#)
- clients/upslog.c, clients/upsmon.c, clients/upsstats.c: Update multi-server clients to use per-connection SSL contexts [\[#3439\]](#)
 - upsmon.c, upslog.c, upsstats.c: - Before connecting to each server, fetch per-server authconf with upscli_get_authconf_item() and create SSL context with upscli_get_or_create_ssl_context_authconf()
 - Attach context to connection via upscli_set_ssl_context() before connect - This allows simultaneous connections to different servers with different client certificates/CAs, resolving the limitation in [issue #3494](#) - Update FIXME comments to NOTE as the limitation is now resolved Benefits: - upsmon can monitor UPS devices behind different CAs/client certs simultaneously - upslog can log multiple systems with independent SSL configurations - upsstats can query multiple data servers with per-server SSL setup - All within a single process, no shared global SSL context constraint
- clients/upsclient.c, clients/upsclient.h: clients/upsclient.h: Expose opaque upscli_ssl_context_config_t type in public header for clarity [\[#3439\]](#) - Add forward declaration of upscli_ssl_context_config_t (struct upscli_ssl_context_config_s) - Change UPSCONN_t::ssl_ctx from void* to typed upscli_ssl_context_config_t* on all platforms - Clarify in comments that ssl_ctx is used for both NSS and OpenSSL (per-connection client cert identity) - Keep extra_reserved padding in non-OpenSSL builds for struct size compatibility - This makes the public API clearer without exposing internals: users know ssl_ctx is a registry handle

- `clients/upsclient.c`, `clients/upsclient.h`: `clients/upsclient.{c,h}`: Complete registry-based SSL multi-context refactor with secure memory zeroing [#3439] - Implement process-wide SSL context registry for caching and reuse - Registry stores config parameters (`certverify`, `certpath`, `certname`, `certpasswd`, `certfile`) - Per-connection references use opaque handles to registry entries (not owned) - NSS per-connection client cert identity now works correctly via registry - Remove `sql_ctx_owned` field (registry handles lifecycle) - Remove per-connection `certident_name/certident_pass` (moved to registry) - Implement secure zeroing of sensitive fields (passwords, cert names/paths) - Registry automatically freed in `upscli_cleanup()` - `upscli_sslinit()` now uses registry context instead of global `ssl_ctx` - Full backward compatibility with legacy `upscli_init*()` APIs via default context - Supports simultaneous multi-realm connections with different CAs and client certs Co-authored-by: GitHub CoPilot
- `clients/upsclient.h`: document current return values for `upscli_init*()` methods [#3439]
- `docs/SSL_CONTEXT_ANALYSIS.adoc`: document the revisited multi-context implementation [#3439]
- `clients/upsclient.c`, `clients/upsclient.h`: `clients/upsclient.{c,h}`: track desired CERTIDENT name/pass via `UPSCONN_t` [#3439]
- `clients/upsclient.c`: `GetClientAuthData()`: pass `UPSCONN_t arg` to use desired cert with its password [#3439]
- `clients/upsclient.c`: `nss_password_callback()`: prefer cert password stored for the connection (if any) over global default (first known pass) [#3439]
- `clients/nutclient.cpp`: `nss_error()`: pass `sock` to use desired cert with its password [#3439]
- `docs/SSL_CONTEXT_ANALYSIS.adoc`: follow-up discussion for true multi-context approach [#3439]
- `docs/SSL_CONTEXT_ANALYSIS.adoc`: add Copilot analysis of current situation in our and ecosystem code [#3439]
- `tests/nutclienttest.cpp`: fix `NutClientTest::test_ssl_context_registry()` to use pure C++ implementation [#3439]
- `clients/nutclient.cpp`, `clients/nutclient.h`: `clients/nutclient.{h,cpp}`: add `Socket` and `TcpClient` getters and setters for `SSLContext` [#3439]
- `NEWS.adoc`: mention *metasys* driver fixed UPS serial number parsing [GHSA-6p5c-r2jg-r9mj]
- `NEWS.adoc`: mention update of *bestups* driver with Megatec Q1 parser fix [GHSA-hwqr-5pjj-3m25]
- `drivers/bestups.c`: update comments and markup [GHSA-hwqr-5pjj-3m25]
- `docs/nut.dict`: add *quiesced* [#3599]
- `docs/developers.txt`: use repetitive pragma combinations via header files [#3599]

2.94 2026-08-30 nmbro <nmbro@users.noreply.github.com>

- `NEWS.adoc`: document *apcmicrolink* driver updates [PR #3591]
- `drivers/apcmicrolink-usb.c`: `fix(apcmicrolink)`: resolve conflicting switch-default warnings `libusb_transfer_status` already had every enumerator listed as an explicit case, so a trailing default was dead code under Clang's `-Wcovered-switch-default` (`-Werror`). Dropping the default then broke the *fightwain* build instead: `-Wswitch-default` demands a default on every switch regardless of enumeration coverage. The two rules cannot both be satisfied by the source text, so suppress `-Wswitch-default` for this switch via the guarded-pragma pattern already used in `apcmicrolink.c`, matching `m4/ax_c_pragmas.m4`'s per-warning detection.
- `docs/Makefile.am`, `docs/apcmicrolink-descriptors.txt`, `docs/new-drivers.txt`: `docs(apcmicrolink)`: document the descriptor format the driver parses Adds `docs/apcmicrolink-descriptors.txt`, covering the frame/page/descriptor structure this driver parses: frames as `[page id][width data bytes][2 byte checksum]`, the flat-blob assembly from page `N` at offset `N * width`, and the descriptor table that maps this wire structure to descriptor paths
 - themselves MIB-like, with attribute IDs that are scope-relative (the same `.11` means "test result" whether it hangs off the battery, UPS, or another collection). Everything in it was derived from a live APC SCL500RMI1UC, for anyone extending `drivers/apcmicrolink-maps.c` to cover more of a device. Wired into the developer guide's build via `docs/Makefile.am` and `docs/new-drivers.txt`. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.

- `drivers/apcmicrolink-maps.c`, `drivers/apcmicrolink.c`, `drivers/apcmicrolink.h`: `fix(apcmicrolink)`: burst-poll a full pass so auth and page reads keep up Root cause: this driver fetched one tunnel record per pollinterval. The device answers requests roughly every 2-3 ms when asked continuously (as PowerChute does), but gates its live-measurement pages behind a slave-password handshake it does not acknowledge until about 20 further exchanges after the response goes out. At one record per pollinterval the handshake's acknowledgement was never collected, and the driver's own readiness check treated sending the response as "done" - so every measurement read 0 indefinitely on a device that was answering normally the whole time. `microlink_poll_burst()` now drives the tunnel like APC's own client: keep requesting the next record until the device stops answering, once per `upsdrv_updateinfo()`, budgeted to one full pass over `page0.count` (bounded by `MLINK_POLL_BURST_MIN/MAX` and an `MLINK_POLL_BURST_MAX_SEC` wall-clock ceiling so a slow-answering device cannot hold `updateinfo()` open indefinitely). The burst also keeps going past its budget while a handshake is in flight (`microlink_auth_pending()`, `MLINK_AUTH_GRA`) so a session re-established mid-`updateinfo()` does not send its auth response on the burst's last record and then go quiet before collecting the answer. `microlink_check_auth_result()` diagnoses the handshake explicitly: `experimental.microlink.diag.auth_status` / `.auth_refused`, an hourly `LOG_WARNING` while refused, and a one-time warning if `AUTH_STATUS` ever sets a bit this driver does not know about. Readiness still accepts `authentication_sent` on its own once the grace window closes - a refused handshake still leaves identity data and the standard-HID fallback worth publishing. `upsdrv_cleanup()` now sends `STOP` before closing the session. Seen live on an `SCL500RMI1UC`: exiting mid-burst left the device answering a later client's `INIT` with whatever page its cursor had reached instead of page 0, and it would not resync until re-enumerated. Also, a batch of descriptor-map corrections gathered while chasing the above (each verified against a live `SCL500RMI1UC`, several by watching PowerChute write or read the same usage): * 2:4.7.28 and 2:4.7.49 are percentages of nominal, not absolute power - they were mapped straight onto `ups.realpower/ups.power`, so both read two orders of magnitude low and `ups.load` then divided one of them by its nominal rating a second time. 2:4.7.28 now feeds `ups.load` directly, and `microlink_publish_derived_power()/microlink_publish_scaled_percent()` derive `ups.realpower` and `ups.power` (the apparent-power side kept as `experimental.ups.load.apparent`, since NUT has no standard name for it) by scaling against the nominal ratings, matching PowerChute's own numbers. * `ups.test.result` now comes from 2:4.5.11 (the battery-scope test, which actually stepped Pending → InProgress → Passed during a PowerChute-triggered self test) rather than 2:11 (UPS-scope, never moved off None on this hardware). 2:11 stays mapped under an experimental name and `microlink_publish_test_result()` promotes it to `ups.test.result` on any device that lacks the battery-scope usage - descriptor attribute IDs are scope-relative, so a differently built model could populate the other one. * 2:4.5.18 is a self-test **schedule** enum, not the interval-in-seconds `ups.test.interval` calls for (two of its members have no interval at all). It is now `experimental.microlink.battery.test.schedule`, and `microlink_publish_test_interval()` derives `ups.test.interval` from the members that do imply a recurring period, matched on the raw bits rather than the label text. * New mappings: `input.sensitivity` (3:25, values confirmed against PowerChute's dropdown), `ups.beeper.status` (2:4.B.3A, both values round-tripped by writing them from PowerChute), `experimental.battery.firmware` (2:4.5.9.4A) and `experimental.statistics.battery.transfers` (2:4.5.F.59, PowerChute's "Number Of Times On Battery"). * Dropped the duplicate device-status publish at 2:4.A: it used the same `apc_status_map` already consumed via `microlink_desc_publish_map` into `ups.status/alarms`, so the second copy was strictly worse (missing LB and the charger flags) rather than a distinct value. * Renamed for consistency: `experimental.device.sku` and `experimental.battery.sku` to `device.part` / `experimental.battery.part` (they are part numbers, not SKUs); the `microlink.*` diagnostic namespace to `experimental.microlink.*` throughout, since none of it is a settled name; and 2:4.9.42 from `experimental.device.sku` to `device.part` directly (an existing standard name this driver had not picked up). Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.

2.95 2026-08-30 Lucas Bocchi <lucas.bocchi@gmail.com>

- `drivers/nhs_ser.c`: Simplify NHS initialization packet helper Pass the packet data, byte count and serial descriptor directly to the helper. Keep serial validation, reopening, write checks and diagnostics inside the helper so the initialization calls are easier to read.
- `NEWS.adoc`, `drivers/nhs_ser.c`: Refresh NHS hardware discovery during driver lifecycle
- `NEWS.adoc`, `docs/nut.dict`, `drivers/nhs_ser.c`: Apply NHS hardware feedback from freechurros Check the `HWINFO` checksum in its interpreter and publish the documented lower-case battery charger states. Log already-known `HWINFO` packets quietly and expose rejected frames through raw packet debugging. Credit the packet captures and follow-up analysis contributed in [issue #3592](#) directly beside the affected code and in the release notes. Suggested-by: Free Churro <freechurrosnet@gmail.com>
- `docs/nut.dict`, `drivers/nhs_ser.c`: Fix strict NHS driver CI diagnostics Initialize protocol and bypass metadata explicitly for every known model so Clang's missing-field check remains clean. Add a documented defensive default to the initialization selector and teach the documentation spell checker the hexadecimal xFE token.

- NEWS.adoc, drivers/nhs_ser.c: Fix NHS serial packet framing and model handling Honor the packet length declared at byte 1 so embedded 0xFF and 0xFE payload bytes do not restart or truncate valid frames. Preserve partial packet state across serial-read timeouts and validate the declared length and final marker before dispatch. Add the compatibility initialization request, model description override, protocol metadata and per-model bypass alarm behavior. Correct the published input voltage extrema and document the changes.

2.96 2026-08-30 Jim Klimov <jimklimov+nut@gmail.com>

- tools/asciidoc-github-links.pl: do not match standalone plus (asciidoc paragraph glue) on a line after a link [#1953]
- tools/Makefile.am: add check-asciidoc-github-links goal [#1953]
- docs/Makefile.am: handle possible absence of perl [#1953]
- tools/asciidoc-github-links.pl: do not re-evaluate patterns in links generated earlier [#1953]
- NEWS.adoc, docs/nut.dict, tools/asciidoc-github-links.pl: tools/asciidoc-github-links.pl, NEWS.adoc: handle lists started with plural "PRs, issues, advisories" [#1953, #3594]
- tools/asciidoc-github-links.pl: use qr / . . . / to define regexes [#1953]
- tools/asciidoc-github-links.pl: streamline with separate definitions of GH URI parts [#1953] This way the script can be more easily reused by other projects and is easier to maintain with symbolic names for repetitive sequences.
- docs/Makefile.am, tools/Makefile.am, tools/asciidoc-github-links.pl: tools/asciidoc-github-links.pl: DOCBUILD_FILTER_GITHUB transplant the stack of Makefile inline SED expressions into a more flexible separate perl script [#1953]
- docs/Makefile.am: *.adoc-parsed: add support for GHSA links [#3594]
- drivers/apcmicrolink-usb.c: new clang insists on casting *malloc() results [#3591]
- .github/workflows/01-make-dist.yml: shell-print debug info using shell variable expansion from envvars Avoid potential code injection via crafted PR source git branch names [GHSA-3xvj-5pxf-f3hg]
- NEWS.adoc: Fix aspell complaint regarding re-synchronization [#3591]

2.97 2026-08-29 user01010111 <lapses.50.booster@icloud.com>

- tests/Makefile.am, tests/cppunit-warnings-end.h, tests/cppunit-warnings.h, tests/cpptest-client.cpp, tests/cpptest.cpp, test-s/example.cpp, tests/nutclienttest.cpp, tests/nutconf_parser_ut.cpp, tests/nutconf_ut.cpp, tests/nutipc_ut.cpp, tests/nutstream_ut.cpp: tests: consolidate CppUnit warning suppression Move the repeated CppUnit diagnostic setup and teardown into paired test-local headers. Keep the zero-as-null-pointer suppression local to cpptest.cpp, correct the Clang push and ignored pragma pair, and distribute the new headers. AI assistance: OpenAI Codex gpt-5.6-sol was used for planning, repository analysis, implementation, review, drafting and macOS validation. OpenAI Codex gpt-5.6-terra was used for Raspberry Pi validation. Both used high reasoning where applicable. The contributor reviewed the change and remains responsible for it.

2.98 2026-08-29 nmbro <nmbro@users.noreply.github.com>

- drivers/apcmicrolink.c: fix(apcmicrolink): use the HID fallback when it is needed The standard-HID-PDC fallback existed but could not engage in the case it was written for. Found on an SCL500RM11UC that answered every poll while reporting an all-zero state: ups.status stayed empty for over two days with usable HID PDC reports arriving on the same endpoint the whole time, and upsmon therefore had nothing to act on. Four separate reasons it never took over, each fixed here: * microlink_start_session() refreshed last_poll_success on a bare handshake, so a device that answered the handshake but sent no data looked freshly polled forever. Data freshness is now tracked separately, and only real polled frames advance it. * upsdrrv_updateinfo() treated a successful reconnect as good data. A handshake proves the device answers, not that the tunnel delivers. * Startup called fatalx() when no fallback snapshot had been decoded yet, even where the device clearly exposes the usages - they simply had not arrived in the startup window. It now starts and publishes them when they do. The old message also claimed no fallback was available on a device whose descriptor advertised one. * Staleness alone was not enough: this

device answers on time and reports zeroes, so the data is fresh and useless. A poll that yields no ups.status flag at all now hands over too. `MLINK_HID_FALLBACK_MAX_AGE_SEC` goes from 10s to 30s. A usbmon capture showed the two streams are not concurrent: PDC reports arrive every 6.0s while the tunnel is idle, then stop for 19.2s whenever the device services tunnel traffic. A 10s window expired inside that gap, so the fallback became unpublishable exactly when the tunnel was also producing nothing, and ups.status went empty once per cycle. `MLINK_DATA_STALE_SEC` moves to 45s to stay above it. Handover is deliberately asymmetric. The fallback takes over immediately but only hands back after the Microlink source has looked plausible for 30s continuously; without that the two swapped on alternate polls, measured at 56 handovers in 3 minutes, and a status that flips every 2s is worse than one that lags a recovery. Plausibility is judged only from ups.status, which is rewritten every poll - testing battery.charge made the fallback read back a value it had published itself and hand over to a dead source. `microlink_publish_hid_fallback()` now writes its measurements before committing status, and states the charging condition explicitly. `dstate's status_commit()` infers `CHRG/DISCHRG` from battery.charge movement when the driver reports neither, and it was comparing the degenerate Microlink value against the previous fallback one and synthesizing a `DISCHRG` that contradicted the `OL` being set in the same breath. Handovers are logged with their reason at debug level 1, once per transition. A device that answers normally while reporting nothing usable also warns at `LOG_WARNING`, hourly, and publishes `microlink.diag.status_degenerate`: a USB bus reset does not clear that state (tested, along with `deauthorize` and `driver_unbind` - none re-enumerate the device), so the user needs to know the UPS itself likely needs a power cycle. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.

- `drivers/apcmicrolink-usb.c`: `fix(apcmicrolink)`: name the async transfer states `CodeQL` flagged the switch on transfer→status as handling only two of the enum's values by name. Behaviour is unchanged - everything not cancelled or gone still falls through to the resubmit below - but the remaining states are now listed explicitly so the intent is readable and a future libusb addition shows up as a warning rather than silently joining the default. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.
- `drivers/apcmicrolink.c`: `feat(apcmicrolink)`: warn on unimplemented stuffing Page 0 bit 1 (`MLINK_PAGE0_FLAG_IMPLICIT_STUFFING`) asks for implicit byte stuffing on the wire. Nothing in the parser implements it; the bit was only published as `microlink.flag.implicit_stuffing` and otherwise ignored, so a device that set it would have its frames read as if stuffing were disabled and would fail with checksum errors that point nowhere near the actual cause. No device seen so far sets the bit - both an `SCL500RMI1UC` and an `SMX1500RM2U` report flags `0x09`, i.e. `AUTH_REQUIRED` and `DESCRIPTOR_PRESENT` with stuffing clear - so rather than ship an unstuffing routine that cannot be exercised or tested, say plainly what the driver is doing and ask for a report. Logged once per process at `LOG_WARNING`, since it is invisible at the default debug level otherwise and describes a condition the user cannot work around. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.
- `NEWS.adoc`, `drivers/apcmicrolink.c`: `fix(apcmicrolink)`: reject a contradictory page 0 Page 0 announces the frame width that every later frame is parsed with, and `microlink_cache_object()` accepted any checksum-valid copy of it. A corrupt copy is therefore unrecoverable: the bogus width becomes the length `microlink_try_extract_frame_at()` demands, no frame checksum-validates again, and the session dies with the device having done nothing wrong. Seen live on a Smart-UPS X 1500 (FW 03.8) in [issue #3587](#). The device served 60 populated pages, then emitted STOP-filled pages and restarted its page index. The wrapped page 0 was checksum-valid and announced a width of 247 inside a 16-byte frame, which replaced the correct width=16/pages=154 header decoded a second earlier. From that point the parser needed 250 contiguous valid bytes out of 19-byte records, so it never extracted another frame. A genuine page 0 always arrives in a frame of exactly the width it announces: on the first read the frame length is derived from that byte, and on every later read the frame length is the established width. So require the two to agree, and keep the previous page 0 when they do not. This is in the shared framing layer, so it applies to the serial transport as well. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.

2.99 2026-08-28 nmbro <nmbro@users.noreply.github.com>

- `NEWS.adoc`, `docs/man/apcmicrolink.txt`, `drivers/apcmicrolink-maps.c`, `drivers/apcmicrolink-usb.c`, `drivers/apcmicrolink-usb.h`, `drivers/apcmicrolink.c`, `drivers/apcmicrolink.h`, `drivers/libusb1.c`, `drivers/nut_libusb.h`: `fix(apcmicrolink)`: async USB reads and stability Long-run testing of the USB HID tunnel transport against an `SCL500RMI1UC` found the driver dropping the session after an inconsistent interval and then failing to recover. Several distinct causes, fixed together here: * The interrupt-IN read was issued per call, so the driver only listened while it happened to be blocked waiting for a reply. This device can go several seconds between replies, and pushes unrelated HID reports on the same pipe meanwhile, so replies arriving outside that window were lost. Reads are now served by a permanently outstanding asynchronous transfer serviced by a dedicated pump thread; only that thread touches libusb's event loop, and everything else takes a lock to drain the queue it fills. Builds without libusb-1.0 or pthreads, and any failure to start the listener, still use the previous synchronous per-call read. This needs

the libusb-1.0 context, exposed as `nut_libusb_get_context()`. * A USB bus reset was triggered off a retry count, which fired just as often for a live-but-stalled device that a reset never helped. It is now triggered only when a read, write or transfer completion has actually reported `NO_DEVICE`, i.e. a genuine unplug or power cycle. * The cached page 0 contents were wiped on every session (re)start, and the read queue was flushed on every session retry - the latter discarding genuine replies that had already arrived. Neither is done any more. * The authentication challenge was a fixed `0x00 0x00`, making the exchange trivially predictable; it is now randomized, as APC's own client does. * When the kernel's `usbhid` driver reclaims the interface, every submission fails instantly with `EBUSY` and nothing paced the retries - one observed incident spun at roughly 14,000 failed submissions per second for over half an hour. That case now backs off and logs, rate-limited. * Warn once when built against libusb-0.1, whose interrupt-IN read did not honor its timeout on at least one tested system, hanging the driver. Also maps two further descriptor usages that were previously visible only as `microlink.unmapped.*`: `experimental.battery.serial` (the battery pack's own serial, confirmed against the compartment label on a real unit) and `microlink.diag.slave_password_echo` (the register the auth challenge is written to, echoed back - useful for troubleshooting, but not by itself proof that authentication was accepted). The `apcmicrolink(8)` USB MODE section is updated to match: it described a periodic reset attempt whenever the tunnel stayed unresponsive, which is no longer what happens, and now also notes the `usbhid/EBUSY` backoff. It gains a "libusb backend" subsection recording that USB mode is tested only against libusb-1.0, what goes wrong on libusb-0.1, and that the always-listening reader needs libusb-1.0 plus `pthread`s. Comments in `apcmicrolink-usb.c` were trimmed throughout: design history, references to material outside this repository, and restatements of what the code already says are gone. Use of coding helper tools and AI disclosed: Claude Code was used for development assistance.

2.100 2026-08-28 Jim Klimov <jimklimov+nut@gmail.com>

- `server/netssl.c`: `ssl_init()`: debug-log the list of enabled OpenSSL ciphers - wrap into pragmas against warnings for OpenSSL headers on some systems [#3331]

2.101 2026-08-27 Jim Klimov <jimklimov+nut@gmail.com>

- `scripts/python/module/PyNUT.py.in`: adapt to lack of overlapping default SSL ciphers [#3331]
- `NEWS.adoc`: made `hidparser.c` `UsageTab` size dynamic [#3588]
- `include/nut_float.h`, `tests/ecoflow_cdc_protocol_utest.c`: define `f_equal_1e_2()` etc. with higher tolerance (1e-2) [#3569]

2.102 2026-08-27 Adam Bedard <adam.bedard@gmail.com>

- `drivers/hidparser.c`, `drivers/hidtypes.h`: `drivers/hidparser.c`: size the HID usage stack from the report descriptor length Follow-up to #3586, where the question was raised whether a real device could legitimately report more than `USAGE_TAB_SIZE` usages and get its descriptor truncated by the bound added there. It could. The usage stack was a fixed `HIDNode_t[50]` inside `HIDParser_t`, and once it filled, parsing stopped and every item after that point was dropped. Before #3586 the same descriptor walked off the end of the allocation instead, so the truncation is new, but the 50-usage ceiling is not - it just used to be enforced by a log line after the damage was done. `Parse_ReportDesc()` already knows the descriptor length, so use it. Every Usage item costs at least the one prefix byte it is introduced by, so an n-byte descriptor can push at most n usages, and an n-slot stack can always hold all of them. The ceiling is gone for any descriptor that fits in the length we were given. `USAGE_TAB_SIZE` stays as the floor so short descriptors keep the capacity they have always had, and the overflow check stays as a backstop - it is now unreachable for a descriptor that fits in n, but it costs one comparison and the invariant it guards is worth stating in code. `UsageSize` goes from `uint8_t` to `size_t`. At its old width it wrapped at 256 regardless of how large the table was, so widening the table alone would not have raised the effective limit. Cost is one `calloc` of 4 bytes per descriptor byte, freed before `Parse_ReportDesc()` returns: at most 24kB for the 6144-byte `MAX_REPORT_SIZE` the libusb and libshut backends cap at, and at most 8kB for the 2048-byte buffer `apc_modbus` gets from `libmodbus`. It happens once per device open, not in the data update loop.
- `drivers/hidparser.c`: refuse truncated HID items instead of reading past the descriptor The item loop checks that `Pos` is inside the descriptor before it reads an item prefix, but nothing checks that the 1, 2 or 4 payload bytes the prefix promises are there as well. A descriptor whose last byte is a size-bearing prefix therefore reads up to 4 bytes past the end of the buffer the caller handed us. Where that lands depends on the backend, and in every case today it is a short read just off the end of a fixed buffer rather than anything wilder: `libusb1` / `libusb0` / `libshut` `rdbuf` is a `MAX_REPORT_SIZE` (6144) byte stack array and `rden` is

capped at `sizeof(rdbuf)`, so the read leaves the array only for `rdlen` in [6141, 6144]. `apc_modbus` the buffer is `libmodbus`'s 2048 byte stack array in `_modbus_rtu_usb_connect()` and the length is the actual transfer count, so the equivalent window is [2045, 2048]. Both windows are device-supplied, so both are reachable, and either way the parser should not read past the length it was given. Refuse the truncated item and stop, which is the same handling the other malformed-descriptor cases get. Reported at debug level rather than `LOG_ERR`: a descriptor that ends mid-item is malformed, but a truncated USB transfer is a plausible way for a real device to produce one, so it belongs with the `PATH_SIZE` refusals rather than with the two that mean malformed or hostile.

2.103 2026-08-26 Jim Klimov <jimklimov+nut@gmail.com>

- `server/netssl.c`: `ssl_init()`: debug-log the list of enabled OpenSSL ciphers [#3331]
- `drivers/microsol-apc.c`, `drivers/upscode2.c`: `drivers/upscode2.c`: revise use of `fabs()` $\Rightarrow$ `fabsf()` for floats
- `tests/ecoflow_cdc_protocol_ustest.c`: revise use of `fabs()` $\Rightarrow$ `f_equal()` [#3569]

2.104 2026-08-26 Adam Bedard <adam.bedard@gmail.com>

- `drivers/hidparser.c`: log HID path depth refusals at debug level [#3586] The five refusals the two prior commits added were all `LOG_ERR`, which is right for the two that only a malformed or hostile descriptor produces - the usage stack overflow and the unbalanced End Collection - but heavy for the three `PATH_SIZE` ones. A path deeper than ten collections is the one case here with a plausible innocent reading: a real device nested further than the parser was built for. Those three drop to `upsdebugx(1)`. The refusal itself is unchanged; only where it is reported moves.

2.105 2026-08-25 Jim Klimov <jimklimov+nut@gmail.com>

- `NEWS.adoc`: document `hidparser` bound-usage and path stacks [#3586]

2.106 2026-08-25 Adam Bedard <adam.bedard@gmail.com>

- `NEWS.adoc`: reword to clear the aspell spellcheck [#3586] The dist tarball job runs aspell over `NEWS.adoc` against `docs/nut.dict`, which has "parsers" but not the possessive "parser's", so the previous wording failed the spellcheck. Reword instead of growing the dictionary; the meaning is unchanged.
- `NEWS.adoc`: clarify what triggers the `hidparser` overflow [#3586] The entry said "too many descriptors", which reads as a count of report descriptors. The overflow needs a single report descriptor carrying more usage items than `USAGE_TAB_SIZE` holds; a device with several descriptors is fine as long as no individual one is over-long.
- `drivers/hidparser.c`: bound the HID path stack and reject unbalanced End Collection Same shape as the usage stack: the `PATH_SIZE` check was a `upslogx()` after the loop rather than a test at the store. The three places that append to `pParser->Data.Path.Nodes` (`ITEM_COLLECTION`, the index node in `ITEM_FEATURE` and friends, and the main-item path store) all wrote first and complained later. `ITEM_END_COLLECTION` had a second problem: it decremented `Path.Size` unconditionally, so an End Collection with no matching Collection wrapped the unsigned counter to 255 and the "remove index if any" test on the next line read far past `Path.Node[PATH_SIZE]`. Unlike the usage stack this one stays inside the `HIDParser_t` allocation, so a sanitizer does not flag it, which is why it is folded in here rather than reported separately. It is not only an overread: on an unbalanced End Collection the unpatched parser returns success with wrong paths - the first node of every path missing and `pathsize` 2 where it should be 3 - after logging "HID path too long". A device reporting silently wrong values is harder to notice than a crash.
- `drivers/hidparser.c`: bound the HID usage stack to fix a heap out-of-bounds write [GHSA-f444-v6f2-2vcj] `ITEM_USAGE` stored into `pParser->UsageTab[pParser->UsageSize]` without ever checking `UsageSize` against `USAGE_TAB_SIZE`. The only `USAGE_TAB_SIZE` check in the file sits after the parsing loop and just calls `upslogx()`, so it reports the problem once the writes have already happened. `UsageTab` is `HIDNode_t[50]` and it is the last member of the heap-allocated `HIDParser_t`, so a report descriptor carrying more than 50 usage items writes attacker-supplied 32-bit values past the end of the allocation. Check before the store instead and return -1, which is the established "stop here" return in this parser: `Parse_ReportDesc()` breaks out

of its loop on `ret < 0`, so everything parsed up to that point is kept and the rest of the descriptor is dropped. Letting `UsageSize` reach `USAGE_TAB_SIZE` exposes a one-past-the-end read in the two unstack loops, which shifted `UsageTab[j+1]` down for every `j < UsageSize`. The last slot has no successor, so shift in 0 there; unused slots are zeroed by `ResetLocalState()` anyway. Reproducer and a before/after test are in the linked advisory. Reachable from a malicious USB device or compromised UPS firmware by way of the report descriptor read in `nut_libusb_open()`.

2.107 2026-08-24 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: fix exclusion for freebsd-15 C89 [#3583]
- Jenkinsfile-dynamatrix: skip failing platform combo for FreeBSD 15 and strict C89 for now [#3583]

2.108 2026-08-23 Jim Klimov <jimklimov+nut@gmail.com>

- tests/.gitignore: add ecoflow-cdc-protocol* file names [#3569]
- common/snprintf.c, include/proto.h: refine check for header availability; indent preprocessor macros per style guide [#3577]
- tests/Makefile.am: link fixreportdesctest-triplite-hid against libm [#3581]
- tests/fixreportdesctest-triplite-hid.c: fix "duplicate declaration" of mock disable_fix_report_desc [#3581]
- drivers/triplite-hid.c: triplite_chemistry_fun(): fix for the case of unresolved model [#3581] At least when included into tests, this code fails on some static analyzers: In file included from fixreportdesctest-triplite-hid.c:60: ../drivers/triplite-hid.c: In function 'triplite_chemistry_fun': ../drivers/triplite-hid.c:179:14: error: argument 1 null where non-null expected [-Werror=nonnull] 179 \ if (!strcmp(model, "1003")) { |~~~~~ In file included from fixreportdesctest-triplite-hid.c:35: /usr/include/string.h:156:12: note: in function 'triplite_chemistry_fun', inlined from 'triplite_chemistry_fun' at ../drivers/triplite-hid.c:171:20: ../drivers/triplite-hid.c:184:14: error: argument 1 null where non-null expected [-Werror=nonnull] 184 \ if (!strcmp(model, "2005")) { |~~~~~ /usr/include/string.h: In function 'triplite_chemistry_fun': /usr/include/string.h:156:12: note: in a call to function 'strcmp' declared 'nonnull' 156 \ extern int strcmp(const char*, const char*) cc1: all warnings being treated as errors
- NEWS.adoc: refer to issue and PR for triplite-hid realpower mapping fix [#3581]
- NEWS.adoc: refer to PR which updated apcmicrolink driver with USB HID support [#3579]
- NEWS.adoc: refer to PR that added omron subdriver [#3554]

2.109 2026-08-22 nmbro <nmbro@users.noreply.github.com>

- drivers/apcmicrolink.c, m4/ax_c_pragmas.m4: fix(apcmicrolink): guard -Wswitch-default/-Wswitch-enum pragmas with feature checks Some compilers lack these specific pragmas even when other diagnostic pragmas are supported; add ax_c_pragmas.m4 detection and guard the unconditional uses in apcmicrolink.c accordingly. Co-Authored-By: Claude Sonnet 5 <noreply@anthropic.com>

2.110 2026-08-21 Jim Klimov <jimklimov+nut@gmail.com>

- tests/fixreportdesctest-triplite-hid.c: fix compiler warnings about static-or-not variable [#3581]
- tests/NIT/nit.sh: log failed test cases with same pattern as gmake errors, for easier search in build logs [#1711]

2.111 2026-08-20 nmbro <nmbro@users.noreply.github.com>

- .gitattributes, .gitignore, NEWS.adoc, data/driver.list.in, docs/man/apcmicrolink.txt, drivers/apcmicrolink.c, tools/nut-usbinfo.pl: refactor(apcmicrolink): use port=auto for USB mode instead of porttype=usb Replace non-standard porttype=usb with standard NUT convention port=auto for USB device selection. Matches nutdrv_qx and other USB drivers. Changes: - USB mode now triggered by port=auto or any USB matching option (vendorid, productid, vendor, product, serial, bus, device) - Removed porttype config option entirely - Updated all documentation and examples - Added proper switched outlet group counting - Minor time() call optimizations in updateinfo - Fixed nut-usbinfo.pl to recognize apcmicrolink-usb.c BREAKING CHANGE: porttype=usb no longer recognized. Use port=auto instead.

- `drivers/apcmicrolink-usb.c`: `fix(apcmicrolink)`: use `usb_reset()` abstraction instead of `libusb_reset_device()` Direct call to `libusb_reset_device()` breaks NUT_USB_VARIANT=0.1 builds. Use `usb_reset()` wrapper from `usb-common.h` which resolves to correct function per USB library version (`libusb_reset_device` on 1.0, `usb_reset` on 0.1).
- `NEWS.adoc`, `data/cmdvartab`, `data/driver.list.in`, `docs/man/apcmicrolink.txt`, `docs/nut-names.txt`, `docs/nut.dict`, `drivers/Makefile.am`, `drivers/apcmicrolink-usb.c`, `drivers/apcmicrolink-usb.h`, `drivers/apcmicrolink.c`: `feat(apcmicrolink)`: add USB HID transport with HID-PDC fallback Tunnel the Microlink protocol over USB HID interrupt IN/OUT for models with no serial port (`porttype = usb`). Tested against a single APC Smart-UPS 500 (SCL500RMI1UC); experimental even relative to the rest of this already-experimental driver. - Session startup on this hardware is intermittently unreliable and can take 90s\+ to respond; refusing to start on handshake failure was the top "driver won't start" complaint. If the device also exposes standard HID Power Device usages, start on that fallback data (`ups.status/battery.charge/battery.runtime`) instead, and bring up outlet-group data/commands automatically once the Microlink tunnel connects, no restart needed. `hid_fallback=no` opts out. - USB read timeout raised to 1s (vs 100ms serial) and handshake retries raised to 10: this device's interrupt pipe interleaves unrelated Input reports with Microlink replies, and a 100ms/3 retry budget starves the handshake before real data arrives. - Add `load.cycle` and per-outlet-group instant commands (`outlet.group.N.load.`, `outlet.group.N.shutdown.`), and `outlet.group.N.switchable` to mark the unswitched main bank (group 0) as having no switching commands.
- The kernel's `usbhid` driver claims this device by default and NUT doesn't detach it; document required `udev` rule in the man page.

2.112 2026-08-20 Kyle Mason <masonkr@gmail.com>

- `tests/.gitignore`, `tests/Makefile.am`, `tests/fixreportdesctest-tripplite-hid.c`: `tests`: cover the `tripplite-hid` Report Descriptor repair Add `fixreportdesctest-tripplite-hid`, modelled on the existing `getexponenttest-belkin-hid`, exercising `tripplite_fix_report_desc()` against synthetic descriptor items so the repair can be verified without the physical UPS attached. Covers the malformed `ActivePower` case proven on a SMART1500LCD, a conformant descriptor that must be left alone, a matching usage without the PDC power unit, the power unit on an unrelated usage, and `disable_fix_report_desc`. Three negative cases document the deliberate scope boundaries: `ApparentPower` and voltage usages are not repaired, and the repair never triggers on value magnitude alone. Note that `get_unit_expo()` and the `HIDUnits` table are both static to `libhid.c` and cannot be linked from a unit test without a live USB handle, so the test names the expected exponent as a constant and asserts the repair's decision and resulting exponent rather than re-deriving `libhid`'s arithmetic.
- `NEWS.adoc`, `docs/nut.dict`, `drivers/tripplite-hid.c`: `drivers/tripplite-hid.c`: recover output power reading, published as 0.0 Two defects combined to make `ups.power` read a constant 0.0 on the tested Tripp Lite SMART1500LCD (09ae:2012) under real load. First, the HID `ActivePower` usages were mapped to `ups.power`. `docs/nut-names.txt` defines `ups.power` as apparent power (VA) and `ups.realpower` as real power (W), so the subdriver published a watts reading under a VA variable name and never published `ups.realpower` at all. The `cps-hid` and `mge-hid` subdrivers already map `ActivePower` to `ups.realpower`. Second, that unit declares its `ActivePower` item with the HID PDC unit for watts/VA (0x0000D121) but leaves the Unit Exponent at 0, while the report already carries a plain integer in watts. NUT's `HIDUnits` table normalizes that unit against an exponent of 7, so `get_unit_expo()` computes 0 - 7 and scales the value by 1e-7. A genuine 520 W reading arrives as raw 0x0208 and is published as 5.2e-05, which the `%.1f` format renders as a confident "0.0". Map both `ActivePower` usages to `ups.realpower`, map `ups.power` to the corresponding `ApparentPower` usages, add `ups.realpower.nominal` from `UPS.Flow.ConfigActivePower`, and add `tripplite_fix_report_desc()` to supply the omitted exponent. The descriptor repair is limited to the `ActivePower` item, matching on `Unit == 0x0000D121 && UnitExp == 0`. No Tripp Lite descriptor is known to expose an `ApparentPower` item, so repairing that usage would be speculative; it remains a mapping target only. A conformant descriptor is untouched, and `disable_fix_report_desc` is honored. Also add `ApparentPower` to `docs/nut.dict`, since the NEWS entry names it. Verified on a SMART1500LCD (09ae:2012): `ups.realpower` now reads 512.0 at `ups.load 56`, consistent with the unit's 900 W rating.

2.113 2026-08-20 Jun Kurihara <junkurihara@users.noreply.github.com>

- `NEWS.adoc`, `drivers/blazer_usb.c`: `blazer_usb`: bound the remaining USB reply scans by the received length Same defect as the one just fixed in the corresponding `nutdrv_qx` transports, in the code those were forked from. `cypress_command()` and `phoenix_command()` fill the caller's buffer in 8-byte chunks under `"i <= buflen - 8"`, so the last iteration can consume the final byte and leave no NUL, after which the `"read: %.*s"` debug line scanned on with an unbounded `strncpy()`. The precision now comes from `memchr(buf, '\r', i)`, falling back to `i`, the accumulated write offset both functions already return. Each block sits behind an explicit `"nut_debug_level >= 3"` check mirroring the `upsdebugx()` macro's own guard, so the new scan too stays off

the normal, non-debugging path. An embedded NUL still stops "%.s" by itself, so a well-formed reply displays as before. That scan sits inside upsdebugx(), a macro guarded by nut_debug_level, so it was only reachable with debugging at level 3 or above, and neither function derives its return value from it. The loop conditions in both functions also scanned the raw receive buffer with strchr(buf, '\r'). Those become memchr(buf, '\r', buflen), which is what the nutdrv_qx counterparts already do. This one is consistency rather than a fix: the condition short-circuits on "i <= buflen - 8", so the unbounded strchr() was never reached with a completely full buffer. DRIVER_VERSION is left alone: it was already bumped for the Ippon fix in this same series. The NEWS entry is widened to name the transports actually covered. A harness includes this translation unit and calls both functions over mocked USB transfers, with the device returning bytes that contain neither CR nor NUL until the buffer is full; its debug stub formats for real, so an oversized precision would be caught too. Against the unpatched tree AddressSanitizer reports a heap-buffer-overflow read in strcspn() for both, at debug level 3 or above and not below it. Both pass after the change, as does the existing Ippon harness, the build with GCC at -Wall -Wextra -Wsign-compare -pedantic -Werror, make check and make spellcheck. No hardware was involved and I have no such unit, so this rests on review and the sanitizers. Coding helper tools were used here: OpenAI Codex and Anthropic Claude Code. The result was reviewed line by line.

- NEWS.adoc, drivers/nutdrv_qx.c: nutdrv_qx: bound the remaining USB reply scans by the received length The Cypress, Phoenix, Fuji and Gtec transports fill the caller's buffer in chunks, guarded by "i <= buflen - chunk", so the final iteration can write its last byte and leave no NUL behind. The trailing scans then had nothing to stop them. Two cases, with different reach. fuji_command() returned strlen(buf) to its caller, which runs unconditionally, so a reply that filled the buffer with neither a CR nor a NUL both read past the end and reported a length derived from whatever followed. It now returns strlen(buf, i), which keeps the padding semantics the comment describes (Fuji pads 8-byte chunks with 0x00) while it cannot run past what was received. strlen() needs no new portability work here: configure.ac probes for it, common/strlen.c is the fallback, and this file already uses it. The other case is the "read: %.s" debug line in all four transports, whose precision came from an unbounded strcspn(). That sits inside upsdebugx(), which is a macro guarded by nut_debug_level, so it was only evaluated with debugging at level 3 or above, and the functions return the accumulated write offset rather than a string length, so nothing outside the debug path consumed the bad value. The precision now comes from memchr(buf, '\r', i) and falls back to i, the same offset the function already returns. Each block sits behind an explicit "nut_debug_level >= 3" check mirroring the upsdebugx() macro's own guard, so the new scan too stays off the normal, non-debugging path. An embedded NUL still stops "%.s" by itself, so the text shown for a well-formed reply does not change. DRIVER_VERSION is left alone: it was already bumped for the Ippon fix in this same series. The NEWS entry is widened to name the transports actually covered. A harness per transport includes this translation unit and calls the real static function over mocked USB transfers, with the device returning bytes that contain neither CR nor NUL until the buffer is full. Its debug stub formats its arguments for real, so an oversized precision would be caught as well as the scan computing it. Against the unpatched tree, AddressSanitizer reports a heap-buffer-overflow read in every one: fuji at debug level 0, and cypress, phoenix and gtec only at level 3 or above, which matches the split above. All pass after the change, as do the existing Ippon harness, the build with GCC at -Wall -Wextra -Wsign-compare -pedantic -Werror, make check and make spellcheck. No hardware was involved and I have none of these units, so this rests on review and the sanitizers. Coding helper tools were used here: OpenAI Codex and Anthropic Claude Code. The result was reviewed line by line.

2.114 2026-08-20 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: For Strict-C builds, skip old GCC on OpenBSD [#3577]
- NEWS.adoc, clients/upsmon.c, common/parseconf.c, common/snprintf.c, common/state.c, common/str.c, configure.ac, docs/nut.dict, drivers/dstate.c, include/common.h, include/proto.h, tests/test_authconf.c, tools/nut-scanner/nut-scanner.c: configure.ac, *.c, *.h: for varargs consider both stdarg.h and sys/stdarg.h [#3577]

2.115 2026-08-19 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upslog.c: only upscli_splitname() if we do not know the answer already
- clients/upslog.c: free the monhost_ups_anchor based list items
- clients/upslog.c: free contents of monhost_ups_anchor based list items
- clients/upslog.c: just in case upscli_disconnect() even for asterisky cleanup
- clients/upslog.c: address valgrind findings for flist and logtarget, and secondary logformat_allocated

- drivers/dummy-ups.c: do not xstrdup() and forget the arg to upscli_upslog_setproctag() [#3379] The setproctag() behind it makes a copy already, so the callers may free() theirs if they like.
- scripts/valgrind/.valgrind.suppress.in: suppress another OpenSSL pattern
- server/upsd.c: upsd_cleanup(): free more of what we allocated [#3329]
- scripts/valgrind/valgrind.sh.in: speed up process startup (do not check/download debuginfo files) Thanks to <https://bbs.archlinux.org/viewtopic.php?id=279297> for the hints.
- server/upsd.c: client_disconnect(): avoid re-closing sock_fd twice
- clients/upsclient.c: upscli_cleanup(), upscli_disconnect(): free more of what we allocated [#3329]
- Makefile.am, tests/Makefile.am, tests/NIT/Makefile.am, tests/NIT/nit.sh: tests/NIT/nit.sh, */Makefile.am: add support for running make memcheck-NIT{-sandbox, }{-devel, } attempts
- tests/Makefile.am: add aliases for check-NIT-sandbox{,-devel}
- Makefile.am: introduce memcheck-quick shortcut
- Makefile.am: rephrase pre-set DISTCHECK_VALGRIND_FLAGS
- clients/authconf.c: handle_authconf_args(): be more diligent about xstrdup() cleanup [#3329]
- docs/configure.txt: clarify inconsistency of NUT --with/--enable flags vs autotools semantics
- tests/test_authconf.c: call upscli_cleanup() to pass memcheck [#3329]
- clients/upsclient.c: upscli_cleanup(): track specifically nss_initialized to avoid teardown when we never initialized NSS (this causes the library to use some memory and fail memcheck)
- ci_build.sh: default-all-errors: report "[Matrix] Error..." for other FAILED scenarios (configure, *clean)
- Jenkinsfile-dynamatrix: in "Strict-C..." filter qw had strict C excluded - bring it back [#3567]
- drivers/main.c: comment on poll_interval sleep location
- clients/upsclient.c, clients/upslog.c, clients/upsstats.c, common/timegm_fallback.c, drivers/generic_gpio_libgpiod.c, drivers/nutdrv_qxpert.c: */.c: fix comments to C89 block style [#3567] Pass at least GCC builds

2.116 2026-08-18 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: rephrase dynacfgPipeline.excludeCombos_AUTOTOOLS_REQUIRE_TWEAK definition as a nested collection [#3567]
- Jenkinsfile-dynamatrix: "Strict-C and GNU C89/C\+\+98 standard builds with fatal warnings (auto level) and GCC toolkits" filter does not actually select Strict-C - be more explicit about this in code/comments [#3567]
- m4/nut_arg_with.m4, configure.ac, NEWS.adoc: revise NUT_ARG* methods to remember nut_withGiven_*/nut_enableGiven_* status of the option, and fix passing one default and one custom user/name value [#3513, #3140]

2.117 2026-08-17 user01010111 <lapses.50.booster@icloud.com>

- drivers/ecoflow-cdc-protocol.c, tests/ecoflow_cdc_protocol_utest.c: Decode EcoFlow telemetry floats portably

2.118 2026-08-17 Costa Tsaousis <costa@netdata.cloud>

- docs/nut.dict: add PR3000ELCDSL/SXL to the spellcheck dictionary The NEWS entry for the cps-hid logical-maximum fix references the CyberPower PR3000ELCDSL and PR3000ELCDSXL models; aspell flags ELCDSL/ELCDSXL as unknown words and fails the NUT spellcheck and distcheck stages on the PR CI. Add both model-name fragments to the accepted words list (alphabetically, next to ELCD).
- NEWS.adoc, drivers/cps-hid.c, tests/getvaluetest.c: drivers/cps-hid.c: fix clipped input and battery voltage on CyberPower 0764:0601 PR3000ELCDSL/PR3000ELCDSXL and sibling models sharing USB ID 0764:0601 declare HID Logical Maximums below the values they transmit for UPS.Input.Voltage/ConfigVoltage (LogMax 70) and UPS.PowerSummary.Voltage/ConfigVoltage (LogMax 255). GetValue() masks with hibit(LogMax) and clamps to [LogMin..LogMax], so input.voltage is reported as 70 and battery.voltage as 22.4 instead of the 225 / 48 V values actually sent on the wire. Extend cps_fix_report_desc() to widen the under-declared maximums using the patterns already established for this device family: - input voltage: also trigger when the HVT LogMax is greater than the input voltage LogMax (oracle sub-form, as used in apc-hid.c), placed as a sibling of the output voltage fix under the HVT block - input config voltage: set LogMax to 255 when the HVT LogMax is greater (byte-sized item; mirrors apc-hid.c) - battery voltage / battery config voltage: raise to CPS_BATTVOLT_LOGMAX when declared below it (constant sub-form, as used for the nominal power fix) All new fix-ups skip items with an active physical scaling (logical_to_physical() would rescale their already-in-range values), stay gated to the existing CPS_VENDORID / 0x0501 / 0x0601 product list, and remain skipable via disable_fix_report_desc. Add getvaluetest cases locking the mask/clamp semantics the fix relies on (wire values 225/230/480 against declared LogMax 70/255 and the widened 511/255/4096). Tested on a PR3000ELCDSXL (0764:0601, firmware CR01901A5P1): before the change input.voltage 70 / battery.voltage 22.4; after 227 / 48.0, with output.voltage and input transfer limits unchanged. [issue #3089]

2.119 2026-08-16 Jim Klimov <jimklimov+nut@gmail.com>

- conf/upsmon.conf.sample.in, docs/man/upsmon.txt, docs/nut.dict: docs/man/upsmon.txt, conf/upsmon.conf.sample.in: document SUSPEND_TIMEJUMP_UNEXPECTED and update other suggestions [#3405]
- drivers/apc_modbus.c: EMBBADSLAVE is not defined in older modbus.h releases [#3571]

2.120 2026-08-15 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: move BUILD_TYPE requirements back from labels to env [#3567]

2.121 2026-08-14 user01010111 <lapses.50.booster@icloud.com>

- tests/Makefile.am: Keep EcoFlow test dependencies local

2.122 2026-08-14 d01 <d01@users.noreply.github.com>

- drivers/apc_modbus.c: apc_modbus: log a read failure once it is one, not once per attempt The retry loop from #3414 kept the pre-retry code's LOG_ERR inside the new loop, so every attempt is logged as an error — including the ones the very next retry recovers, which is the exact case the loop was added to handle. On a device that needs retries routinely this floods the system log with errors the driver does not have. Measured on an APC Smart-UPS X1500 in a degraded state: ~310 LOG_ERR/min while the connection was never closed once and every read ultimately succeeded — about 890k lines/day and 42% of the journal, doubled again by upsdrvctl forwarding the driver's stderr alongside its own syslog write. Log attempts with upsdebugx(1), and the error once, after the loop, where the read has really failed and the connection is about to be closed. Both lines carry the attempt count. Nothing is logged when exit_flag broke the loop: the driver is stopping, not failing. Co-Authored-By: Claude Fable 5 <noreply@anthropic.com>
- drivers/apc_modbus.c: apc_modbus: retry reads whose reply belongs to another request The retry loop from #3414 retries only on ETIMEDOUT and gives up on every other error. That leaves out the one case where a retry is most obviously correct: a reply arrived, but it is not the reply to this request. libmodbus reports those as EMBBADSLAVE, EMBBADCRC or EMBBADDATA. On a packetised transport such as rtu_usb this is common, because a reply the device deferred past our

timeout stays queued on the endpoint rather than ageing off the wire as it would on an idle serial line. The next read then collects the previous request's answer. Reading it has consumed it, so the very next attempt can succeed—but the loop breaks out instead of making it. On an APC Smart-UPS X1500 this is exactly what happens on the first driver start after a host reboot (the self-powered UPS keeps its queue across the host's power cycle): the initial read times out, the retry collects the stale reply, and the loop gives up, so startup fails with "Can't read inventory information from the UPS". systemd then restarts the driver, and each failed start abandons further exchanges, degrading the device until only reseating the USB cable recovers it. Genuine Modbus exception responses still break out of the loop immediately: the device answering properly will not improve on a retry. Co-Authored-By: Claude Fable 5 <noreply@anthropic.com>

- drivers/apc_modbus.c: apc_modbus: stop retrying reads once exit_flag is set The read retry loop added in #3414 does not check exit_flag, so a driver asked to stop keeps working through its remaining retries: up to retries * response_timeout, which with non-default settings comfortably exceeds the 5 seconds upsdrvctl allows for SIGTERM before it escalates to SIGKILL (drivers/upsdrvctl.c, the "retrying harder" path). On some hardware that escalation is not merely untidy: an APC Smart-UPS X1500 (051d:0003, FW "UPS 16.0") killed mid-exchange stops serving Modbus altogether—it accepts writes and answers nothing—and only physically reseating the USB cable recovers it. Every clean shutdown then wedges the UPS. Break out of the loop as soon as exit_flag is set. Verified on that hardware: shutdowns now log "Signal 15: exiting" and deactivate cleanly where they previously hit "Stopping ... failed, retrying harder" followed by status=9/KILL. Co-Authored-By: Claude Fable 5 <noreply@anthropic.com>

2.123 2026-08-14 Jim Klimov <jimklimov@gmail.com>

- NEWS.adoc: update the note about snmp-ups voltronic subdriver [#3570]

2.124 2026-08-14 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: introduce dynacfgPipeline.excludeCombos_AUTOTOOLS_NOT_REQUIRE_TWEAK and dedup "manpage docs builds" vs. "autotools builds" [#3567]
- Jenkinsfile-dynamatrix: define excludeCombos_AUTOTOOLS_REQUIRE_TWEAK to simplify rule sets maintenance [#3567]
- Jenkinsfile-dynamatrix: comment on "all/manpage docs types" vs. "autotools-driven builds" [#3567]
- Jenkinsfile-dynamatrix: update labels (non-docs), comment on dedup potential of "all driver types on capable systems" vs. "combos with Python versions for helper scripts" [#3567]
- Jenkinsfile-dynamatrix: run "Various GNU C99+/C++11+ standard builds with fatal warnings (auto level), without distcheck and docs" scenario also out-of-dir [#3567]
- Jenkinsfile-dynamatrix: dedup: "Various non-docs non-distchecked target builds with main and \~newest supported C/C++ revisions" (default-nodoc) is nowadays a subset of several default-all-errors scenarios [#3567]
- Jenkinsfile-dynamatrix: dedup: limit "Various GNU C99+/C++11+ standard builds with fatal warnings (auto level), without distcheck and docs" scenario to non-GCC toolkits [#3567] They ran essentially same workload twice, as "Various GNU C99+/C++98+ standard out-of-tree builds with fatal warnings (hard level) with GCC"
- ci_build.sh: comment about DISTCHECK_TGT="distcheck-ci" in BUILD_TYPE="default-all-errors"* handler [#3567] Defined just in case, but currently not used (DAE does not do distcheck now).
- Jenkinsfile-dynamatrix: annotate build scenarios currently used for master branch and common PR builds [#3567]
- NEWS.adoc: document the three fixes for apc_modbus [#3571]
- drivers/apc_modbus.c: bump DRIVER_VERSION, small style fixes [#3571]
- docs/nut.dict: docs/nut.,dict: add PSCERT entry [#3570]

2.125 2026-08-13 user01010111 <lapses.50.booster@icloud.com>

- drivers/Makefile.am, drivers/apc-hid.c, drivers/arduino-hid.c, drivers/belkin-hid.c, drivers/cps-hid.c, drivers/delta_ups-hid.c, drivers/ecoflow-cdc.h, drivers/{ecoflow-cdc.c ⇒ ecoflow-hid-aux-cdc.c}, drivers/ecoflow-hid-aux-cdc.h, drivers/ecoflow-hid.c, drivers/ever-hid.c, drivers/explore-hid.c, drivers/idowell-hid.c, drivers/legrand-hid.c, drivers/liebert-hid.c, drivers/mge-hid.c, drivers/openups-hid.c, drivers/powercom-hid.c, drivers/powervar-hid.c, drivers/salicru-hid.c, drivers/tripplite-hid.c, drivers/usbhid-ups.c, tests/ecoflow_cdc_protocol_utest.c: Address HID auxiliary review feedback
- drivers/Makefile.am: Fix Windows static link order for EcoFlow CDC
- docs/nut.dict: Accept EcoFlow option name in documentation spellcheck
- drivers/Makefile.am: Include EcoFlow CDC headers in source tarballs
- docs/man/usbhid-ups.txt, drivers/Makefile.am, drivers/apc-hid.c, drivers/arduino-hid.c, drivers/belkin-hid.c, drivers/cps-hid.c, drivers/delta_ups-hid.c, drivers/ecoflow-cdc-protocol.c, drivers/ecoflow-cdc-protocol.h, drivers/ecoflow-cdc.c, drivers/ecoflow-cdc.h, drivers/ecoflow-hid.c, drivers/ever-hid.c, drivers/explore-hid.c, drivers/idowell-hid.c, drivers/legrand-hid.c, drivers/liebert-hid.c, drivers/mge-hid.c, drivers/openups-hid.c, drivers/powercom-hid.c, drivers/powervar-hid.c, drivers/salicru-hid.c, drivers/tripplite-hid.c, drivers/usbhid-ups.c, drivers/usbhid-ups.h, tests/Makefile.am, tests/ecoflow_cdc_protocol_utest.c: Add optional EcoFlow CDC telemetry to usbhid-ups
- NEWS.adoc, drivers/libusb0.c, drivers/libusb1.c: Prefer complete HID descriptor for EcoFlow devices

2.126 2026-08-13 Jun Kurihara <junkurihara@users.noreply.github.com>

- NEWS.adoc, drivers/blazer_usb.c: blazer_usb: bound Ippon USB reply parsing by the number of bytes read The legacy Ippon transport carries the same defect just fixed in the nutdrv_qx Ippon subdriver, which was forked from this code. It reads up to 64 bytes into a 64-byte stack buffer with usb_interrupt_read() and then found the end of the reply with strcspn(), unbounded by the transfer length, so a reply filling the buffer without a CR or a NUL scanned past the end of the array. The length that came back drove snprintf("%.*s", len, tmp) into the caller's buffer, so bytes adjacent on the stack could be handed back as reply data. The driver is obsoleted, but it is still built and shipped. The fix is not a copy of the nutdrv_qx one, because the two functions do not agree on where a reply ends. strcspn(tmp, "\r") stops at whichever of CR or NUL comes first, which is why the Ippon empty-reply frame 0x00 0x0D yields 0 here while nutdrv_qx yields 2. The bounded replacement therefore takes the earlier of memchr(tmp, '\r', ret) and memchr(tmp, '\0', ret), and falls back to ret when the reply contains neither. The following "\v+1 to include the terminator" step is kept, but skipped when no terminator was received: there is nothing to count in that case, and incrementing would hand snprintf() a precision of 65 against a 64-byte array holding no NUL, reintroducing the overread it is meant to remove. Behavior is otherwise unchanged, including two quirks left alone deliberately rather than corrected inside this change. The returned length counts a terminating NUL that snprintf() did not copy, and it is not clamped when the destination truncates. blazer.c only compares that value against thresholds and treats the buffer as a C string, so neither is harmful. DRIVER_VERSION is bumped to 0.27 and NEWS.adoc records the change. No Ippon hardware was available, so this rests on review and on a local harness that calls the real ippon_command() over mocked USB transfers. The harness pinned the four behaviors above against the unpatched tree first, and they are unchanged after the fix. Under AddressSanitizer the old code reports a stack-buffer-overflow read of 65 bytes at the strcspn() call with nut_debug_level at 0, confirming the overread is not confined to a debug path; the patched code is clean at debug levels 0 and 5. Built warning-free with GCC at -Wall -Wextra -Wsign-compare -pedantic -Werror. Coding helper tools were used here: OpenAI Codex and Anthropic Claude Code. The result was reviewed line by line.
- NEWS.adoc, drivers/nutdrv_qx.c: nutdrv_qx: bound Ippon USB reply parsing by the number of bytes read ippon_command() reads up to 64 bytes into a 64-byte stack buffer with usb_interrupt_read(). The end of the reply was then located with strlen() and strcspn(), neither of which is bounded by the transfer length, so a reply filling the buffer without containing a CR or a NUL made both scan past the end of the array. The resulting length was not merely logged: it reached the memcpy() into the caller's buffer and the returned length, so bytes adjacent to the buffer on the stack could be handed back as reply data. Take the count returned by usb_interrupt_read() as the authoritative extent instead. A CR within those bytes ends the reply as before, CR included. Failing that, a NUL within them ends it, which is the bounded equivalent of the old strlen() fallback. Failing both, every received byte is payload. Hex logging, text logging and the copy all use that bounded length, and the text log strips a trailing CR with an explicit length check rather than strcspn(). Behavior is otherwise unchanged: truncation to buflen - 1, NUL termination of the destination, and appending a missing terminating CR when there is room. One consequence is intended. When the device replied with fewer bytes than the command just written from the same buffer, the old strlen() fallback could

run into the leftover command bytes and count them as reply data; the bounded scan cannot. DRIVER_VERSION is bumped to 0.55 and NEWS.adoc records the change. Coding helper tools were used here: OpenAI Codex and Anthropic Claude Code. The result was reviewed line by line, built warning-free with GCC at -Wall -Wextra -Wsign-compare -pedantic -Werror, and exercised with a local harness that calls the real `ippon_command()` over mocked USB transfers under AddressSanitizer and UndefinedBehaviorSanitizer.

2.127 2026-08-13 Jim Klimov <jimklimov+nut@gmail.com>

- docs/developers.txt: revise sections on indentation, line breaks, etc. [#3569]
- Jenkinsfile-dynamatrix: make slowBuild definitions more similar - move BUILD_TYPE into dynamatrixAxesVirtualLabelsMap everywhere [#3567]
- NEWS.adoc, docs/nut.dict, drivers/snmp-ups.c: NEWS.adoc, drivers/snmp-ups.c: announce voltronic MIB subdriver, bump snmp-ups version [#3570]

2.128 2026-08-13 Chris Allen <203867+chrisallen@users.noreply.github.com>

- data/driver.list.in, docs/man/snmp-ups.txt, docs/nut.dict, drivers/Makefile.am, drivers/snmp-ups.c, drivers/voltronic-mib.c, drivers/voltronic-mib.h: drivers/snmp-ups: add "voltronic" subdriver for enterprise 43943 UPSes Add support for Voltronic enterprise 43943 SNMP cards, tested with a PowerShield Centurion RT 1000VA and PSSNMPV4 card. The private MIB gives better resolution for most readings, so use it first and keep RFC1628 as a fallback. Direct GETs showed that the private line tables use index 0 while the RFC1628 tables use index 1, despite the MIB declarations not making that difference clear. Map identity, topology, status, battery data, input, output, bypass, ratings, load and measured real power. Publish the canonical aggregate power names alongside the useful output aliases. The card returns zero for apparent power, power factor and the generic transfer limits even when the UPS is carrying load. Leave those values out instead of presenting them as measurements or trying to calculate them from load. Leave the optional PSEMD branch out as well because no connected sensor was available to verify it. Delayed shutdown, startup and beeper controls follow the vendor MIB and firmware definitions. They have not been tested on live hardware, and no control command was sent during development. Add the subdriver to the build, document its limits, and add the tested PowerShield model to the compatibility list.

2.129 2026-08-12 Yuriy_Y <alliancesko@gmail.com>

- NEWS.adoc, data/driver.list.in, docs/man/nutdrv_qx.txt, drivers/nutdrv_qx.c: nutdrv_qx: add richcomm-svc USB transport [#3562] Add an explicit USB communication subdriver for older session-initialized Richcomm USB-to-UART bridges. The tested SVC V-1000F uses the existing Megatec protocol and Armac command framing, but requires a 2400-baud report and a 64-byte session-opening sequence before it accepts commands. Ignore leading empty HID reports from this bridge and reset its transport state after USB reconnection. Use the 6-byte reads required by the V-1000F without changing the 8-byte reads used by normal armac devices. Keep the mode opt-in because 0925:1234 is shared by devices with different communication behavior. Document the mandatory configuration, tested hardware and untested command scope in the man page, HCL and NEWS. Development and documentation were assisted by OpenAI Codex. The submitter reviewed the changes and verified the hardware behavior.

2.130 2026-08-12 Falusi Tamás <tamas.falusi@gmail.com>

- NEWS.adoc, docs/nut.dict, drivers/apc-hid.c: apc-hid: map shutdown.return for APC Back-UPS BX*MI units [#2683] On newer APC Back-UPS devices (e.g. Back-UPS BX750MI, 051d:0002) the only shutdown-capable HID usage is UPS.APCGeneralColl. The previously only matching command mapped to it, shutdown.reboot, writes the default value 10, which this firmware accepts (USB-level success) but never executes — so the built-in shutdown sequence reported success without ever powering the load off. Map shutdown.return to the same usage with value 1, matching the write APC PowerChute Serial Shutdown performs (confirmed by decoding the USBPcap captures archived in [issue #2683](#), and live on a 2025-11 BX750MI-GR unit): the load is cycled after a fixed ~2 minute grace (the value is not a seconds count on this firmware), the command executes only while on battery, and output returns when wall power is present, even if AC came back during the grace period. With this mapping the driver-default shutdown command chain (shutdown.return,shutdown.reboot,load.off.delay,shutdown.stayoff) succeeds on

its first command: verified on battery with `usbhid-ups -k` on a BX750MI-GR — output cut 2 min 2 s after the write and re-energized on AC restore. Co-Authored-By: Claude Fable 5 <noreply@anthropic.com> Claude-Session: https://claude.ai/code/session_01QQEAfDBxtHupEdAcv3Bgz2

2.131 2026-08-11 Jim Klimov <jimklimov+nut@gmail.com>

- docs/nut.dict: Update with new entries [#3563]
- drivers/nutdrv_qx.c: upsdrv_initups(): define `val` for all code paths [#3563]
- docs/man/nutdrv_qx.txt: document `cypress_drain_quirk` setting [#3563]
- UPGRADING.adoc: nutdrv_qx added `cypress_0665_5161_subdriver` method [#3563] Clarified impact of `cypress/cypress_0665_5161_subdriver` fix on deployments and provided workaround instructions.
- NEWS.adoc: nutdrv_qx added `cypress_0665_5161_subdriver` method [#3563] Clarified impact of `cypress/cypress_0665_5161_subdriver` fix on deployments and provided workaround instructions.
- drivers/nutdrv_qx.c: bump version, let `cypress_drain_quirk` be configured via `ups.conf` [#3563]

2.132 2026-08-10 pixi-p <pixip@o2.pl>

- data/driver.list.in, drivers/nutdrv_qx.c: nutdrv_qx: fix Cypress 0665:5161 replies Drain stale interrupt reports before issuing a command for the Cypress 0665:5161 transport, and use a longer reply timeout for this device quirk. This allows Voltronic-QS autodetection for a VOLT Polska RackUPS 1200VA/720W. Related to #2534.

2.133 2026-08-09 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsc.c, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsrw.c, clients/upsset.c, clients/upsstats.c: Honour per-authconf flags `ssl` (do not get stuck on default) [#3331]
- clients/upscli.c: upscli_init2(): avoid potential leak if re-entered [#3331]
- server/upsd.c: log whether incoming data from a CLIENT is in an encrypted or plaintext session [#3331]
- docs/Makefile.am: SOURCE_CHECK_UNICODE: avoid printing useless `SRCDIR=""` in some locations
- clients/authconf.c, clients/authconf.h, clients/upsc.c, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsrw.c, clients/upsset.c, clients/upsstats.c, docs/man/upscli_read_authconf_file.txt, drivers/dummy-ups.c, tests/test_authconf.c, tools/nut-scanner/scan_nut.c: clients/*.* et al: upscli_read_authconf_file(): add an argument to hide best-effort nutauth.conf discovery failure into debug log [#3329]

2.134 2026-08-06 Jim Klimov <jimklimov+nut@gmail.com>

- README.adoc: switch to star-history.dera.page fork for GitHub stars graph See <https://github.com/star-history/star-history/-/issues/539>
 - Jenkinsfile-dynamatrix: update different group names to highlight what they do now, update comments to mark up what can be optimized
 - Jenkinsfile-dynamatrix: revise excludeCombos stacking style to minimize diffs in comparisons
 - Jenkinsfile-dynamatrix: update comment about excluding NetBSD from default autotools builds
 - Jenkinsfile-dynamatrix: update autotools group names to highlight that they run `distcheck-ci` too
 - Jenkinsfile-dynamatrix: silence default `MAKE_OPTS` for `dynacfgPipeline.buildPhases[distcheck]` (autotools)
 - Jenkinsfile-dynamatrix: add `*.adoc` to `dynacfgPipeline.appliesToChangedFilesRegex_FILES_TXT`
-

2.135 2026-08-05 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, docs/FAQ.txt, docs/Makefile.am, docs/developers.txt, docs/man/upsd.conf.txt, docs/security.txt, scripts/Windows/README.adoc, docs/Makefile.am, *.txt, *.adoc: Revise asciidoc markup around monospaced words Various parsers have trouble with closing backtick followed by a character which is not a word separator (dash seems OK, an apostrophe precludes closing the monospace context). Added a spellcheck-like rule to avoid regressions about this, and fixed earlier reporting in that docs/Makefile.am goal.

2.136 2026-08-04 Marcelo Pacheco <marcello.mpacheco@gmail.com>

- NEWS.adoc, common/wincompat.c: win32: fix short reads on serial timeout

2.137 2026-08-04 Jun Kurihara <junkurihara@users.noreply.github.com>

- drivers/nutdrv_qx.c: fix C+\+ compatibility in OMRON transport Cast memchr() results explicitly to char * so C+\+ compatibility checks do not reject the implicit conversion from void *.

2.138 2026-08-04 Jim Klimov <jimklimov+nut@gmail.com>

- README.adoc: Change of star-history URL did not help, revert it For more context, <https://github.com/star-history/star-history/issues/539>
- README.adoc: star-history graph now requires a token GitHub API tightening: * <https://github.blog/changelog/2026-06-30-upcoming-access-restrictions-to-public-api-endpoints-and-ui-views/> * <https://www.star-history.com/blog/github-stargazer-api-restriction/>
- docs/nut.dict: add QTS [#3554]

2.139 2026-08-03 mkiera <itskiera20@gmail.com>

- NEWS.adoc: note the GoldenMate LiFePO4 Logical Maximum fix [PR #3555]
- drivers/idowell-hid.c: idowell-hid: fix bogus battery.runtime/battery.voltage on GoldenMate LiFePO4 GoldenMate 1000VA/800W LiFePO4 packs (06da:ffff, and the same firmware on 075d:0300) declare their data twice: as Feature items in ReportID 0x01 with sane limits, and again as Input items in ReportID 0x02. In ReportID 0x02 the firmware emits the 4-byte Logical Maximum for UPS.PowerSummary.RunTimeToEmpty byte-reversed, as "27 ff ff ff fe" (0xFEFFFFFFF) where 0xFFFFFFFEE was intended: 09 68 Usage (RunTimeToEmpty) 75 20 Report Size (32) 15 00 Logical Minimum (0) 27 ff ff ff fe Logical Maximum (0xFEFFFFFFF = 4278190079) 81 83 Input 0xFEFFFFFFF does not fit a signed 32-bit long, so on LLP64 platforms (Windows, where long is 32-bit) the generic "LogMax < LogMin" recovery in HIDParse() stores it back via "(long) pParser->Value" and it wraps to -16777217. Logical values persist in HID global state, and the UPS.PowerSummary.Voltage Input item that follows declares no Logical Maximum of its own, so it inherits the same broken limit: 05 84 09 30 Usage (Voltage) 67 21 d1 f0 00 Unit 55 05 Unit Exponent (5) 75 10 Report Size (16) 81 82 Input ← no new Logical Maximum Both readings then come out as nonsense (battery.runtime = -16777217, battery.voltage = -167772.2) while the ReportID 0x01 Feature copies of the very same usages read correctly (1740 and 13.2). LP64 platforms are unaffected, since 4278190079 fits a 64-bit long there. Add idowell_fix_report_desc() to repair the two ReportID 0x02 items. Rather than invent limits, use the values this device itself declares for the same usages in ReportID 0x01: 0x75FFFFFF for RunTimeToEmpty (32-bit) and 65535 for Voltage (16-bit). Only items whose maximum is demonstrably broken (below the minimum) are touched, so a firmware revision that encodes this correctly is left alone, and the user's disable_fix_report_desc toggle is honored.

2.140 2026-08-03 Jun Kurihara <junkurihara@users.noreply.github.com>

- docs/man/nutdrv_qx.txt, drivers/nutdrv_qx.c: nutdrv_qx: clarify OMRON USB transport details [#3112] Describe the sub-driver option as selecting a USB communication subdriver so it also covers the OMRON HID transport. Document that OMRON's non-standard SET_REPORT value 0x0002 is intentional, matches both vendor driver releases, and was verified on the BN150T. This changes only the literal's presentation, not its value or behavior.

- docs/nut.dict: docs: update spelling dictionary for OMRON documentation [#3112] Add DSM, OMRON's, QNAP and Synology to the spelling dictionary for terms introduced by the OMRON documentation.
- NEWS.adoc, docs/man/nutdrv_qx.txt, drivers/nutdrv_qx.c, drivers/nutdrv_qx_omron.c: docs: refine OMRON documentation and comments [#3112] Improve the wording of the OMRON NEWS entry, manual section and source comments. Clarify the tested scope, protocol behavior and source provenance while keeping the private report details out of the public documentation. No driver logic is changed.
- NEWS.adoc, data/driver.list.in, docs/man/nutdrv_qx.txt, drivers/nutdrv_qx_omron.c: docs: record OMRON BN150T hardware verification [#3112]
- NEWS.adoc, data/driver.list.in, docs/man/nutdrv_qx.txt, drivers/nutdrv_qx_omron.c: docs: clarify provenance of OMRON GPL driver sources [#3112] Identify the OMRON-authored driver versions as 1.00 from the Synology DSM 7.3-86009 GPL sources and 1.02 from the QNAP QTS 5.2.3 GPL sources. Replace ambiguous references to vendor releases consistently across the manual, NEWS, driver list and source comments.

2.141 2026-07-31 Jun Kurihara <junkurihara@users.noreply.github.com>

- NEWS.adoc, docs/man/nutdrv_qx.txt, drivers/nutdrv_qx.c, drivers/nutdrv_qx_omron.c: nutdrv_qx: add OMRON protocol extensions Add BN150T extension queries, OMRON-specific status handling, and shutdown command encoding and acknowledgement handling. Prefer the vendor-ID-gated OMRON protocol during USB autodetection, and document the tested scope and known limitations.
- NEWS.adoc, data/driver.list.in, docs/man/nutdrv_qx.txt, docs/nut.dict, docs/nutdrv_qx-subdrivers.txt, drivers/nutdrv_qx.c: docs: document the "omron" protocol and USB subdriver [#3112] Adds the OMRON BN150T to the compatibility list at support level 4, the protocol to the nutdrv_qx man page, the subdriver source files to the developer guide's list, and the two words the spell checker needs. The man page states what was actually exercised: the polling path on one model over USB. None of the shutdown commands has been sent to real hardware, so the section says so rather than implying coverage. ondelay is accepted for this protocol but does not reach the device: OMRON's shutdown commands carry no return-delay field.
- drivers/Makefile.am, drivers/nutdrv_qx.c, drivers/nutdrv_qx_omron.c, drivers/nutdrv_qx_omron.h: drivers/nutdrv_qx: add "omron" protocol subdriver for OMRON BN150T [#3112] The BN150T reports the third Q1 status bit as 1 continuously while on mains, idle and unloaded, which "q1" turns into BYPASS - in NUT convention a state where protection has been bypassed. OMRON's own driver does not evaluate that bit at all: the block deciding between TRIM, BYPASS and BOOST is commented out in its entirety, identically in the 1.00 and 1.02 vendor releases, while every other status bit is handled. The row is dropped here rather than remapped, since what the bit reports on this hardware is not known. Instant commands the vendor driver does not implement are not registered: test.battery.start, .deep, .quick and .stop have no T or TL command anywhere in either vendor tree, the vendor's instant command set has no whole-load on/off behind load.on and load.off, and beeper.toggle has no bare Q. Registering an instant command tells clients the device can perform it. The shutdown commands use OMRON's byte strings. The vendor sends "S<n>" and "Sf<n>" with the delay in tenths of a minute below one minute and in whole minutes above, and never appends the Megatec "R<mmmm>" return-delay field; blazer_process_command() appends it to both as soon as ups.delay.start is non-zero, and spells stay-off as "S<n>R0000" where OMRON has a distinct "Sf". shutdown.return and shutdown.stayoff are also two transactions rather than one: the auto-restart flag is set with "An" or "Af" first, as both the vendor driver and docs/nut-names.txt require, and a failure there abandons the command instead of powering the unit down with its restart behaviour unknown. A command the UPS accepts is acknowledged with "OK", not the "ACK" that the Megatec-derived subdrivers expect. The claim is blazer_claim_light() gated on OMRON's USB vendor ID. Without the gate this subdriver, which checks no more than "q1" does, would claim every device the generic q1 fallback exists to serve. A serial-attached unit therefore needs protocol=omron set explicitly. The USB transport handler added earlier is renamed omron_usb_subdriver(), because the subdriver_t structure takes the plain name; same treatment as fabula_hunnox_subdriver(). Verified on 0590:00b7: polling reports OL without the spurious BYPASS. No instant command has been sent to the hardware, so every command string here rests on the vendor source alone. Other OMRON models untested.

2.142 2026-07-30 Jun Kurihara <junkurihara@users.noreply.github.com>

- drivers/nutdrv_qx.c: register OMRON BN150T (0590:00b7) for the "omron" subdriver [#3112] nut-usbinfo.pl generates the udev, hotplug and devd rules from the driver device tables, so without a qx_usb_id[] row a packaged install grants the device no permissions at all. The row needs a matching omron_subdriver(): qx_is_usb_device_supported() only reports SUPPORTED

once a handler has assigned `subdriver_command`. Only the tested model is listed; the other OMRON IDs shipped by the vendor driver are untested.

- `drivers/nutdrv_qx.c`: add "omron" USB subdriver for OMRON BN150T [#3112] Same transport as "ippon", except the control transfer carries a 16-byte HID Output report, the size declared by the BN150T's report descriptor. The send loop advances by the report size and rejects a short transfer, and the reply length and both debug prints are bounded by the number of bytes actually read. Verified on 0590:00b7 with `protocol=q1`. Other OMRON models untested.

2.143 2026-07-30 Jim Klimov <jimklimov+nut@gmail.com>

- `tests/nutclienttest.cpp`: add `NutClientTest::test_ssl_context_registry()` (currently via `upsccli_set_ssl_context()` not `C++`) [#3439]
- `clients/nutclient.cpp`: make OpenSSL "ssl_ctx" and NSS "_nss_initialized" flag properties of the `Socket` class instance, not static globals assigned once per process [#3439]
- `clients/upscliclient.c`, `clients/upscliclient.h`: `clients/upscliclient.{c,h}`: add support for connection-specific SSL context [#3439] ... as an alternative to the global SSL context, which is not suitable for multiple connections with different SSL parameters (certificates, trusted CA realms, etc.) in the same process.
- `NEWS.adoc`: relocate USB `rdlen` fix notice to "fallout from earlier development" section and detail a bit more [#3550]

2.144 2026-07-28 John Grant <john.grant@dronedeploy.com>

- `NEWS.adoc`: note the report descriptor length fix [PR #3550]
- `drivers/libusb0.c`, `drivers/libusb1.c`: `drivers/libusb{0,1}.c`: bound `rdlens` loop by element count, not `sizeof` [#3136] `nut_libusb_open()` iterates the two candidate HID report descriptor lengths: `int32_t rdlen1, rdlen2, rdlens[2]; size_t j; ... for (j = 0; j < sizeof(rdlens); j++)` `sizeof(rdlens)` is 8 (bytes), not 2 (elements), so the loop reads `rdlens[2..7]` - 24 bytes past the end of the array - and uses that stack garbage as `rdlen`. The "ran out of candidates" check after the loop has the same bug. The out-of-range indices are only reached when neither real candidate satisfies the caller, i.e. exactly when a device is in a bad state and both descriptor reads fail. Garbage that happens to pass the `rdlen` sanity checks is then handed to `libusb_control_transfer()` and on to the HID parser callback, which is where it segfaults. Seen with `usbhid-ups` against an EcoFlow DELTA 3 Plus (3746:ffff) on an NVIDIA Jetson Orin Nano whose USB device had wedged; "usbreset" on the device cleared the wedge and hid the crash. Use `SIZEOF_ARRAY()` from `common.h`, which both files already include. `libusb0.c` carries the identical defect and is fixed the same way. Regression from `aba6f43544b17d2f16e69ca19f162c6c19748fc7` ("if the tried `rdlen` did not succeed, fall back to the other value we had in mind"), first released in v2.8.5.

2.145 2026-07-28 Jim Klimov <jimklimov+nut@gmail.com>

- `NEWS.adoc`, `docs/nut.dict`, `tools/nut-scanner/scan_eaton_serial.c`, `tools/nut-scanner/scan_ipmi.c`, `tools/nut-scanner/scan_nut.c`, `tools/nut-scanner/scan_snmp.c`, `tools/nut-scanner/scan_xml_http.c`: `tools/nut-scanner/scan_*.c`, `NEWS.adoc`, `docs/nut.dict`: fix mutex release in failure code path [#3551]
- `clients/upssched.c`: `conf_arg()`: do not hard-code use of `\\.\pipe\upssched`, only fall back to it [#3525]
- `common/common.c`, `include/common.h`, `clients/upssched.c`, `drivers/dstate.c`, `NEWS.adoc`: introduce `upsdebug_ascii_compact()` for better readability [#3525]
- `drivers/dstate.c`: `send_to_one()/send_to_all()`: report sent buffer content via `upsdebug_ascii()` to avoid confusing trailing empty line for `\n` [#3525]
- `clients/upssched.c`: `send_to_one()`: report sent buffer content via `upsdebug_ascii()` to avoid confusing trailing empty line for `\n` [#3525]

2.146 2026-07-27 Jim Klimov <jimklimov+nut@gmail.com>

- docs/config-prereqs.txt: update for FreeBSD 15 fresh install
- drivers/apcsmart.c, drivers/belkin.c, drivers/bestfortress.c, drivers/genericups.c, drivers/nhs_ser.c, drivers/powerpanel.c, drivers/powervar_cx_usb.c, drivers/tripplite.c, drivers/tripplite_usb.c, drivers/usbhid-ups.c: drivers/*.c: remove useless custom logging, hide useful messages behind may_log_reconnect_trying() [#3541]
- drivers/apcsmart.c: revise tracking of serial port file descriptor [#3541]
- drivers/main.c, drivers/main.h: drivers/main.{c,h}: introduce may_log_reconnect_trying() for drivers that retain custom logging of reconnection details for now [#3541]
- drivers/main.c, conf/ups.conf.sample, docs/man/ups.conf.txt: Introduce reconnect_report_freq to manage verbosity of on-going reconnect_trying() streak [#3541]
- drivers/tripplite_usb.c, UPGRADING.adoc: remove old bespoke reconnection limit tracking [#3541]
- drivers/tripplite.c, NEWS.adoc: retry failed connections [#3541]
- NEWS.adoc: list drivers impacted by reconnect_max_tries in one place [#3541]
- drivers/powervar_cx_usb.c, UPGRADING.adoc: remove old bespoke reconnection limit tracking [#3541]
- drivers/powerpanel.c, NEWS.adoc: retry failed connections [#3541]
- drivers/genericups.c: handle valid/invalid upsfd like in bestfortress.c [#3541]
- drivers/genericups.c, NEWS.adoc: retry failed connections [#3541]
- drivers/belkin.c: handle valid/invalid upsfd like in bestfortress.c [#3541]
- drivers/bestfortress.c, NEWS.adoc: retry failed connections [#3541]
- NEWS.adoc: fix indentation style
- drivers/belkin.c, NEWS.adoc: retry failed connections [#3541]
- clients/upssched.c: update comment about HANDLE passing [#3525]
- clients/upssched.c, NEWS.adoc: rewrite WIN32 timer daemon spawning code path [#3525] Co-authored-by: @alexdrun-cpu Co-authored-by: Claude Sonnet 4.6
- NEWS.adoc: fix indentation typo [#3546]

2.147 2026-07-25 José San Juan <josesj@gmail.com>

- NEWS.adoc, drivers/riello_ser.c, drivers/riello_usb.c: riello_usb, riello_ser: report NAK on RE as failure

2.148 2026-07-24 Lucas Willian Bocchi <lucas@lucas.inf.br>

- NEWS.adoc, docs/man/nhs_ser.txt: Document NHS serial configuration changes
-

2.149 2026-07-24 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, drivers/adelsystem_cbi.c, drivers/apc_modbus.c, drivers/apcsmart.c, drivers/bcmxcp_usb.c, drivers/blazer_usb.c, drivers/generic_modbus.c, drivers/nhs_ser.c, drivers/nutdrv_qx.c, drivers/powerman-pdu.c, drivers/powervar_cx_usb.c, drivers/richcomm_usb.c, drivers/riello_usb.c, drivers/triplite_usb.c, drivers/usbhid-ups.c: drivers/*.c, NEWS.adoc: update drivers to use reconnect_trying() with enum arg values for consistent reporting [#3541]
- drivers/main.c: reconnect_trying(): fix syslog facility usage [#3541]
- drivers/main.c, drivers/main.h: drivers/main.{c,h}: reconnect_trying(): refactor to use reconnect_state_t as enum with symbolic names instead of magic numbers [#3541]
- drivers/main.c, drivers/main.h: drivers/main.{c,h}: accept reconnect_trying(2) to consistently process "reconnect.updateinfo" [#3541]
- drivers/nhs_ser.c: use set_exit_flag() with symbolic EF_EXIT_* values [#2670]
- drivers/main.c, drivers/dstate.c: use set_exit_flag() and symbolic EF_EXIT_* values instead of exit_flag=NUM [#2670] Also compare symbolic not numeric values when deciding on exit code.
- drivers/main.c: set_exit_flag(): use common EF_EXIT_FAILURE and EF_EXIT_SUCCESS symbolic values to log debug traces [#2670, #1923]
- drivers/main.c: set_exit_flag(): log debug traces based on new value, not old [#1923]

2.150 2026-07-23 Lucas Bocchi <lucas.bocchii@gmail.com>

- drivers/nhs_ser.c: Modernize NHS serial communication and add configurable port settings

2.151 2026-07-23 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, conf/ups.conf.sample, docs/man/ups.conf.txt, drivers/main.c, drivers/main.h: drivers/main.{c,h}, docs/man/ups.conf.txt, conf/ups.conf.sample, NEWS.adoc: introduce reconnect_max_tries setting for NUT drivers [#3541]
- clients/nutclient.cpp, scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: clients/upsclient.c, clients/nutclient.cpp, scripts/python/module/PyNUT.py.in, scripts/perl/UPS/Nut.pm.in: be sure to normalize authconf sections with various spellings for user, host and port in the same manner [#3329]
- tests/test_authconf.c: verify that empty hostname is normalized to localhost [#3329]
- scripts/python/module/PyNUT.py.in: findAuthConf(): add optional debug tracing [#3329]
- clients/upsclient.c, clients/nutclient.cpp, scripts/python/module/PyNUT.py.in, scripts/perl/UPS/Nut.pm.in: be consistent about empty hostname in [admin@:12345] being treated as localhost [#3329]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: add support for NUT_DEBUG_LEVEL envvar

2.152 2026-07-22 dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>

- .github/workflows/08-PyNUTClient.yml: build(deps): bump actions/setup-python from 6 to 7 Bumps actions/setup-python from 6 to 7. - Release notes - Commits --- updated-dependencies: - dependency-name: actions/setup-python dependency-version: 7 dependency-type: direct:production update-type: version-update:semver-major ...

2.153 2026-07-22 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: when username gets manually edited, update auto-suggested password accordingly [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in: fix ups_port_entry handler to not block TAB through the field [#3514]

2.154 2026-07-21 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: use the more idiomatic approach to focus order chains [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in: explicitly set initial focus to host field [#3514] It was like that before, but leave nothing to chance :)
- scripts/python/app/NUT-Monitor-py2gtk2.in: __check_gui_fields(): get port value more reliably [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in: update authconf suggestions as the port number is being typed [#3514]
- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/app/NUT-Monitor-py3qt{5,6}.in: set initial focus to host field [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: call self.__check_gui_fields() when ups_port_entry is edited [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: removed the empty gui_status_message() call from __check_gui_fields [#3514] It was causing relevant messages (like mismatches or discovery results) to disappear instantly.
- scripts/python/app/NUT-Monitor-py2gtk2.in: self.refresh_gui() immediately after updating the status message [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: clear the "Use auth" checkbox if no longer suggesting from authconf [#3514]
- drivers/upsdrvctl.c: help(): clarify wording for list <ups> operation [#2567]
- drivers/upsdrvctl.c, NEWS.adoc: start_driver(): we can re-check started instance also for single device runs [#3302]
- drivers/upsdrvctl.c: start_driver(), forkexec(): pepper with debug traces [#3302]

2.155 2026-07-20 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/locale/NUT-Monitor.pot, .../python/app/locale/fr/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/fr/fr.po, .../python/app/locale/it/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/it/it.po, .../python/app/locale/ru/LC_MESSAGES/NUT-Monitor.mo, scripts/python/app/locale/ru/ru.po, scripts/python/app/ui/about-dialog1.ui, scripts/python/app/ui/gui-1.3.glade, scripts/python/app/ui/gui-1.3.glade.h: scripts/python/app/NUT-Monitor-py*.in, NEWS.adoc: bump NUT-Monitor client versions to 2.1.0 [#3514, #3509]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: fix focus chain (TAB/SHIFT-TAB order) of login page interface elements
- docs/man/NUT-Monitor.txt: update with nutauth.conf support [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: revise enablement of login/pass fields in connection with "User auth" button [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: fix real-time update of credential suggestion as host/port are being edited [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: revise ups_authentication_check checkbox for authconf driven suggestions [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: fix tracking of suggested/edited login/pass flag and style as other fields get edited [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: fix code style after manual and AI edits [#3509]

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: if the default authconf section suggested credentials already, highlight that too [#3514]
- scripts/python/module/PyNUT.py.in: readAuthConfFile(): fix handling of section name followed by comment [#3329, #3514]
- scripts/python/module/PyNUT.py.in: refactor with AuthConf.printDebug() and print to stderr [#1711]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: consider entered login/password or unchecked box for use of them when logging in with authconf (for SSL etc) [#3514]
- NEWS.adoc: update for refactor of background() method [#3437]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: refactor handling if user/password fields were suggested vs manually edited (at least one of them) [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: readAuthConfFile() calls: use options to trace absent nutauth.conf files [#3329, #3514]
- scripts/python/module/PyNUT.py.in: readAuthConfFile(): add options to trace absent nutauth.conf files [#3329, #3514]
- scripts/python/module/PyNUT.py.in: add AuthConf.clone() helper method [#3329, #3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: track if user/password fields were suggested (per authconf) and highlight them then [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-py*.in: avoid a RecursionError on console when closing the UI window [#3514] Avoid Error in sys.excepthook: RecursionError: maximum recursion depth exceeded
- drivers/main.c: fix comment style to old C [#3437]
- NEWS.adoc, docs/nut.dict, drivers/nutdrv_qx_voltronic-qs.c: drivers/nutdrv_qx_voltronic-qs.c, NEWS.adoc: bump VOLTRONIC_QS and reword the news entry [#3521]
- drivers/nutdrv_qx_voltronic-qs-hex.c, NEWS.adoc: fix format string for ups.load [#3532]
- INSTALL.nut.adoc: update with a larger note about packages lagging behind development tip [#3531] Also remove references to package maintainers being part of core team, there was a bit of turbulence in the past years, such notes get obsolete. . .

2.156 2026-07-19 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-*.in: fix loading of saved "favorites" passwords (not base64-decoded before) [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-*.in: consider authconf hits for SSL setup [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-*.in: only replace login/pass if never edited by user but rather suggested from authconf [#3514]
- NEWS.adoc, scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/app/NUT-Monitor-py*.in: add support for NUT Auth configuration files [#3514]
- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in, scripts/python/app/NUT-Monitor-*.in: refactor old_handler teardown with new graceful and harsher disconnect() handling [#3509]

- scripts/python/app/NUT-Monitor-py2gtk2.in, scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/module/PyNUT.py.in: scripts/python/app/NUT-Monitor-*.in, scripts/python/module/PyNUT.py.in: refactor with a graceful and harsher disconnect() handling [#3509]
- scripts/python/app/NUT-Monitor-py2gtk2.in: handle lack of pynotify more gracefully [#3509]
- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/app/NUT-Monitor-py3qt{5,6}.in: refactor gui_status_message() with try/except handling [#3509]
- scripts/python/app/NUT-Monitor-py2gtk2.in: gui_status_message(): fix indentation [#3509]
- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/app/NUT-Monitor-py3qt{5,6}.in: fix "Broken pipe" and possible stall in reconnection attempts [#3509]
- scripts/python/module/PyNUT.py.in: refactor with an error-logging __send() [#3509]
- scripts/python/module/PyNUT.py.in: __read_until(): handle the case of un-received chunk [#3509]
- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: scripts/python/app/NUT-Monitor-py3qt{5,6}.in: mark the templates as nominally executable

2.157 2026-07-17 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/mge-hid.c, NEWS.adoc: bump DRIVER_VERSION and update news about CHRG status suppression [#3518, #3519]

2.158 2026-07-15 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, scripts/python/app/NUT-Monitor-py2gtk2.in: scripts/python/app/NUT-Monitor-py2gtk2.in: propose a fix for reconnection modeled after PR #3523 for Python3 siblings [#3509] Co-Authored-By: Programmer <developer.sorli@gmail.com>
- NEWS.adoc: NUT-Monitor updated with automatic reconnection [#3509, #3523]
- NEWS.adoc, drivers/blazer_usb.c, drivers/nutdrv_atcl_usb.c, drivers/nutdrv_qx.c, drivers/powercom-hid.c, drivers/richcomm_usb.c, drivers/riello_usb.c, drivers/tripplite_usb.c: drivers/*.c, NEWS.adoc: play safer with temporary buffers, pre-zero some [#3529]
- clients/upsmon.c: cast assignments into temporary argv[] [#3499]
- tests/NIT/upssched.conf.in: fix path to upssched-cmd sample [#3499, #1711]
- tests/NIT/nit.sh: we no longer support NOTIFYCMD with shell snippets in one token [#3499]
- clients/upsmon.c: upsmon_cleanup(): NULLify freed allocations; free cert-related data [#3499]
- NEWS.adoc, clients/upsmon.c, clients/upssched.c: clients/upssched.c: add an atexit(clean_exit) handler similar to that in upsmon [#3499]
- NEWS.adoc, UPGRADING.adoc, clients/upsmon.c, clients/upssched.c, conf/upsmon.conf.sample.in, conf/upssched.conf.sample.in, docs/man/upsmon.conf.txt, docs/man/upssched.conf.txt: upsmon, upssched: extend SHUTDOWNCMD, NOTIFYMSG and CMDSCRIPT to handle command-line args as separately quoted tokens in the directive [#3499]

2.159 2026-07-14 Jim Klimov <jimklimov+nut@gmail.com>

- common/wincompat.c, drivers/dstate.c, clients/upssched.c: CreateNamedPipe(): add PIPE_REJECT_REMOTE_CLIENTS flag to only work with local host clients [#3527]
 - common/wincompat.c: syslog() shim: protect against re-entry if we upslog()/upsdebug() from this method [#3527]
 - common/wincompat.c: syslog() shim: do not flood stderr with repetitive SKIP: can't open existing event log NAMED_PIPE messages [#3527]
 - clients/upssched.c: help(): fix reference to man page
-

- clients/upssched.c, NEWS.adoc: fix notify_msg from CLI when other options are present [#3525, #3105]
- tests/NIT/nit.sh: do not die() in certificate prerequisite checks if we can follow up, logging the error in a way that seems too fatal to warnings parsers [#1711]
- ci_build.sh, tests/NIT/nit.sh: simplify cut_filter() [#3526, #3109]
- Do not add optional dependency on cut * Consider starting AND ending characters of input hash string for more randomness
- NEWS.adoc, ci_build.sh, tests/NIT/nit.sh: ci_build.sh, tests/NIT/nit.sh: use shorter hash strings for cache dirs [#3526, #3109]

2.160 2026-07-13 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upssched.c, common/wincompat.c, NEWS.adoc: use WIN32 security attributes for listening pipes [#3525] Replicate what was done for dstate.c in PR #3479. Co-authored-by: mbv06 <mbv06.dev@gmail.com>
- common/wincompat.c, include/wincompat.h, drivers/dstate.c: move init_pipe_security() to be reusable [#3479, #3525] Co-authored-by: mbv06 <mbv06.dev@gmail.com>
- include/wincompat.h: move include guard to be after heading comments
- configure.ac: if CPPUNIT seems present but won't build, suggest if C+\+ version could be the problem
- scripts/perl/UPS/Nut.pm.in, scripts/perl/test_nutclient.pl, scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py: fix parsing of CERTIDENT NAME/PASS from nutauth.conf files [#3329]
- clients/authconf.c, clients/nutclient.cpp, scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: consistently log seen unsupported keywords in nutauth.conf files [#3329] ... this can well be a typo in parser code, help catch those.
- tests/NIT/nit.sh: revise generation of PEM files with NSS binary builds [#1733, #3329] Now we need the PEM files always (when we can make them) for PERL and Python test suites. Generation of JKS (e.g. for jNut) remains conditional - if we would maintain the test certificate cache.
- tests/NIT/nit.sh: fall back in nutauth-openssl.conf to use rootca.pem if no CERTHASH.0 links were populated [#1711, #3329]
- tests/NIT/nit.sh: actively not-enable "CERTFILE" in nutauth.conf if upsmon.pem is missing [#1711, #3329]
- tests/NIT/nit.sh: reshuffle determination of DO_USE_NIT_TESTCERT_CACHE and log different outcomes [#1711, #3329]
- tests/NIT/nit.sh: at least report that upsmon.pem is missing and tests may fail, if we did not get one fallbacked for NSS builds [#1711]
- tests/NIT/nit.sh: annotate failure messages for easier search in make-oriented logs [#1711]
- ci_build.sh: annotate default-all-errors failure messages for easier search in make-oriented logs [#1711]

2.161 2026-07-12 Jim Klimov <jimklimov+nut@gmail.com>

- configure.ac: check for systemctl earlier, so it feeds into optional warning about lack of systemdsystemunitdir

2.162 2026-07-09 Jim Klimov <jimklimov+nut@gmail.com>

- clients/Makefile.am, clients/upsclient.c, clients/upsclient.h: clients/upsclient.{c,h}: keep UPSCONN_t size same for different SSL backends [#3331]
-

2.163 2026-07-05 Programer <developer.sorli@gmail.com>

- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: NUT-Monitor: Add automatic reconnection mechanism for Qt5 and Qt6 - Add automatic reconnection when connection to NUT server is lost - Implement proper teardown of old PyNUTClient instances - Use specific exceptions (socket.error, ConnectionError, EOFError, OSError, PyNUT.PyNUTError) - Prevent multiple simultaneous reconnection attempts - Add *was_online* flag to track connection state changes
- Tested and confirmed working on both Qt5 and Qt6 Closes: [#3509](#)

2.164 2026-07-03 Stephen Horvath <s.horvath@outlook.com.au>

- docs/nut.dict: Add voltronic-qs *QI* command
- NEWS.adoc, data/driver.list.in: NEWS.adoc, driver.list.in: Update for voltronic-qs & PSDR800
- drivers/nutdrv_qx_voltronic-qs.c: nutdrv_qx: Voltronic-QS: Add QI command This command adds battery charge, runtime, input frequency, output current, and input transfer low/high values to the Voltronic-QS driver. Tested on a PowerShield PSDR800.

2.165 2026-07-01 strasharo <3337997+strasharo@users.noreply.github.com>

- drivers/mge-hid.c: mge-hid: suppress CHRG on CC-mode devices when battery is fully charged UPS.PowerSummary.PresentStatus.C stays asserted in float/ maintenance mode on constant-charge (CC-mode) devices such as the Eaton 5E series. These devices do not expose the ABM HID path (UPS.BatterySystem.Charger.ABMEnable), so advanced_battery_monitoring stays ABM_UNKNOWN. eaton_abm_check_chrg_fun() then maps any Charging=1 report directly to CHRG with no distinction between active charging and float mode. The result is ups.status stuck at OL CHRG indefinitely, even when battery.charge is 100. Fix: when ABM is UNKNOWN or DISABLED and the UPS asserts Charging=1, check battery.charge before publishing CHRG. If the battery is at 100% the charger is in float/maintenance mode, not actively charging, so return !chrg instead. Tested on Eaton 5E 650i (vendor 0463, product ffff, subdriver MGE HID 1.58, firmware 01.04.0016) where ups.status now correctly shows OL at full charge instead of OL CHRG.

2.166 2026-06-30 Jim Klimov <jimklimov@gmail.com>

- tests/NIT/nit.sh: setenv_ssl_perl(): in case of darwin OS, neuter CERTVERIFY also in nutauth.conf [[#1711](#)]

2.167 2026-06-30 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c, clients/authconf.h: clients/authconf.h: do not require nut_stdint.h in public API [[#3329](#)]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: fix AuthConf::authconf_list to be actually a list, not vector [[#3329](#)] Memory-safer to extend and manipulate.
- clients/nutclient.cpp: SSLConfig::addCertHost(): avoid duplicates [[#3329](#)]
- tests/NIT/nit.sh: setenv_ssl_perl(): fix up paths on Windows in a new nutauth-openssl-perl-win.conf [[#3329](#), [#1711](#)]
- tools/gitlog2changelog.py.in: fix unicode TypeError with python 2.7
- scripts/python/module/PyNUT.py.in: AI added "import getpass" needlessly [[#3329](#)]
- scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in: scripts/python/module/PyNUT.py.in: do not strip brackets from section names after creation (fix IPv6 address handling) [[#3229](#), [#1711](#)]
- scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in: avoid Python 3 formatting strings [[#3229](#), [#1711](#)] There is no reason for module to not work for older OS installations.
- tests/NIT/nit.sh: setenv_ssl_*(?): pass NUT_SSL=true for SSL builds even if skipping other settings due to nutauth.conf [[#3329](#), [#1711](#)] May neuter later in the method (e.g. perl lacking SSL module).
- NEWS.adoc, clients/upsmon.c, clients/upsmon.h, common/nutconf.cpp, common/nutwriter.cpp, include/nutconf.hpp: clients/upsmon. nutconf.cpp, nutwriter.cpp, NEWS.adoc: introduce NOTIFY_SUSPEND_TIMEJUMP_UNEXPECTED [[#3405](#)]

2.168 2026-06-29 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/module/test_nutclient.py.in, scripts/perl/test_nutclient.pl, tests/cpptest-client.cpp: only set "NUT_SSL=True" from AuthConf NUT_FORCESSL if no value is passed as envvar [#3329, #1711]
- tests/cpptest-client.cpp: setUp(): fix env_NUT_CERTHOST_ADDR and env_NUT_CERTHOST_NAME [#3329, #1711]
- tests/cpptest-client.cpp: setUp(): fix env_NUT_CERTSTORE_PATH vs env_NUT_CAPATH (nss vs openssl backend) [#3329, #1711]
- tests/cpptest-client.cpp: setUpClientSSL(): trace incoming envvars and configs being applied [#3329, #1711]
- tests/cpptest-client.cpp: setUp(): fix env_NUT_CERTIDENT_NAME vs env_NUT_KEYFILE [#3329, #1711]
- tests/cpptest-client.cpp: revise certfile vs certpath handling [#3329, #1711]
- scripts/python/module/test_nutclient.py.in: no longer need detailed AuthConf debug in tests [#3329]
- scripts/python/module/PyNUT.py.in: fix readAuthConfFile() to strip spaces from KEY [#3329]
- scripts/python/module/PyNUT.py.in: fix readAuthConfFile() to create new section with stripped brackets [#3329]
- scripts/python/module/PyNUT.py.in: fix readAuthConfFile() to use open() not os.open() [#3329]
- scripts/python/module/PyNUT.py.in: revise logging in AuthConf getter, reader, merger [#3329]
- scripts/python/module/PyNUT.py.in: add PyNUT.AuthConf.str() method [#3329, #1711]
- scripts/perl/test_nutclient.pl: revise certfile vs certpath handling
- tests/NIT/nit.sh: refine NUT_SSL=false vs NUT_IGNORE_AUTHCONF [#1711, #3329]
- tests/NIT/Makefile.am: check-NIT-devel: we now depend on more clients [#1711]

2.169 2026-06-28 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/module/PyNUT.py.in: track AuthConf section name value without the brackets [#1711, #3329]
 - scripts/perl/test_nutclient.pl: tone down AuthConf parser debugging now [#1711, #3329]
 - tests/NIT/nit.sh: populate upsmon.pem for OpenSSL-only clients when generating NSS certs [#1711, #3311]
 - scripts/perl/UPS/Nut.pm.in: fix key casing in readAuthConfFile() [#3329]
 - scripts/perl/UPS/Nut.pm.in: more logging and comments to readAuthConfFile() [#3329]
 - tests/NIT/nit.sh: generatecfg_nutauth(): produce a nutauth-openssl.conf for Perl and Python tests to work even with NSS C builds [#3329, #1711]
 - scripts/perl/UPS/Nut.pm.in: rectify merge() logging [#3329]
 - tests/NIT/nit.sh: generate nutauth.conf with both "localhost" and "127.0.0.1" entries for different queries [#3329, #1711]
 - scripts/perl/UPS/Nut.pm.in, scripts/perl/test_nutclient.pl, scripts/python/module/test_nutclient.py.in: scripts/perl/UPS/Nut.pm.in, scripts/perl/test_nutclient.pl: rectify findAuthConf()/getAuthConf() logic and logging [#3329]
 - docs/Makefile.am: ChangeLog.adoc-parsed: convert markdown links to asciidoc (grok dependabot commit comments) [dependabot/core#15228]
 - NEWS.adoc: move ragtech entry to post-2.8.5 release
-

2.170 2026-06-26 juslex <66561713+juslex@users.noreply.github.com>

- scripts/upsdrvsvctnl/nut-driver-enumerator.sh.in, tests/nut-driver-enumerator-test—ups.conf, tests/nut-driver-enumerator-test.sh: nut-driver-enumerator: classify ragtech as serial media The ragtech driver talks to a USB CDC-ACM device node (/dev/tty-ACM*) through the serial layer. Classifying it as "serial" media lets the auto-generated nut-driver@<ups> service depend on systemd-udev / nut-udev-settle, so it waits for the device node to be published by udev before starting instead of racing the enumeration at boot. Requested by @jimklimov in [issue #3500](#). Add a [ragtech-serial] section to the enumerator self-test fixture and the matching entries in the affected expected outputs (device-name listing, config dump, and MEDIA/checksum listing). Full self-test passes (104 cases).

2.171 2026-06-26 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/module/test_nutclient.py.in: fix interpretation of ac.certfile [[#3329](#)]
- tests/cpptest-client.cpp: debug progress through authconf parsing [[#3329](#), [#1711](#)]
- scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in: scripts/python/module/*: debug progress through authconf parsing
- scripts/perl/UPS/Nut.pm.in, scripts/perl/test_nutclient.pl: scripts/perl/*: debug progress through authconf parsing
- scripts/python/module/PyNUT.py.in: AuthConf: fix upper-/any- case expectations [[#3329](#)]
- scripts/perl/UPS/Nut.pm.in: readAuthConfFile(): fix upper-/any- case expectations [[#3329](#)]
- tests/NIT/nit.sh: report current SSLBACKEND in nutauth.conf [[#1711](#)]
- clients/nutclient.cpp: handle_authconf_args(): AuthConf::merge(): pepper with debug [[#3329](#)]
- clients/nutclient.cpp: handle_authconf_args(): set user name early (if known from section), so we can ignore it in set_authconf_val() [[#3329](#)]
- clients/nutclient.cpp: set_authconf_val(): fix upper-/any- case expectations [[#3329](#)]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{h,cpp}: introduce methods to dump AuthConf data for debug [[#3329](#)]
- clients/authconf.c: update comment [[#3329](#)]
- scripts/python/module/test_nutclient.py.in, scripts/python/module/PyNUT.py.in: reshuffle debugging to see into AuthConf handling [[#3329](#), [#1711](#)]
- scripts/python/module/PyNUT.py.in: debug-log more details about SSL setup if it is enabled [[#3329](#), [#1711](#)]
- clients/authconf.h, clients/nutclient.h, scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: C, C+\+, Perl, Python AuthConf implementations: update comment about "certfile" field [[#3329](#)]
- conf/upsmon.conf.sample.in, docs/man/upsmon.conf.txt: fix CERTFILE description ("client" not "server" here) [[#3329](#)]
- scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in: scripts/python/module/PyNUT.py.in: rename AuthConf "pass_" field to "password" [[#3329](#)]
- scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: document meanings of AuthConf fields [[#3329](#)]
- scripts/python/module/PyNUT.py.in: fix back mis-merged edits [[#3329](#)]
- scripts/perl/test_nutclient.pl: reshuffle debugging to see into AuthConf handling [[#3329](#), [#1711](#)]
- scripts/perl/UPS/Nut.pm.in: UPS::Nut→new(): support args hash with an AUTHCONF key to parse it and merge with other args [[#3329](#)]
- scripts/perl/UPS/Nut.pm.in: revise debug logging of AuthConf progress [[#3329](#)]
- clients/nutclient.cpp: parse_authconf_file(): upsdebugx() which file we are looking at [[#3329](#)]

- scripts/perl/test_nutclient.pl, scripts/python/module/PyNUT.py.in, scripts/python/module/test_nutclient.py.in, tests/cpptest-client.cpp: Fix C++\+, Perl, Python test runners to consider nutauth.conf [#3329] WIP
- scripts/perl/UPS/Nut.pm.in: refactor error/debug messaging from readAuthConfFile() [#3329]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{h,cpp}: add AuthConf operator= [#3329]

2.172 2026-06-25 Thanh Nguyen <imscopilot@outlook.com>

- data/driver.list.in: Add ARES AR630 to HCL

2.173 2026-06-24 dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>

- .github/workflows/01-make-dist-worker.yml, .github/workflows/05-codeql.yml: build(deps): bump actions/cache from 5 to 6 Bumps [actions/cache](#) from 5 to 6. - [Release notes](#) - [Changelog](#)
 - [Commits](#) --- updated-dependencies: - dependency-name: actions/cache dependency-version: 6 dependency-type: direct:production update-type: version-update:semver-major ...
- .github/workflows/01-make-dist-worker.yml, .github/workflows/05-codeql.yml: build(deps): bump actions/checkout from 6 to 7 Bumps [actions/checkout](#) from 6 to 7. - [Release notes](#) - [Changelog](#)
 - [Commits](#) --- updated-dependencies: - dependency-name: actions/checkout dependency-version: 7 dependency-type: direct:production update-type: version-update:semver-major ...

2.174 2026-06-24 Jim Klimov <jimklimov+nut@gmail.com>

- tests/cpptest-client.cpp, scripts/perl/test_nutclient.pl, scripts/python/module/test_nutclient.py.in: use nututh.conf files if possible [#3329, #1711]
- tests/NIT/nit.sh: setenv_ssl_*(): skip work by default if nutauth.conf exists [#1711, #3329]
- scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: revise splitting of host:port strings where host looks like numeric IPv6 [#3503]
- clients/nutclient.cpp: AuthConf::merge(): gracefully handle section name that starts with "@" [#3329]
- clients/nutclient.cpp, scripts/perl/UPS/Nut.pm.in, scripts/python/module/PyNUT.py.in: revise normalization of host strings that look like numeric IPv6 [#3503]
- clients/nutclient.cpp: update splitaddr() against libupsclient variant
- scripts/python/module/test_nutclient.py.in, tests/NIT/Makefile.am: scripts/python/module/test_nutclient.py: add testsuite_splitaddr() [#3329, #1711]
- scripts/python/module/PyNUT.py.in: allow to construct PyNUTClient without connecting right away; remember requested TRACKING option to apply whenever we do connect() [#3329, #1711]
- configure.ac, scripts/Makefile.am, scripts/perl/.gitignore, scripts/perl/UPS/{Nut.pm ⇒ Nut.pm.in}, tests/NIT/Makefile.am, tests/NIT/nit.sh: Convert UPS/Nut.pm into a .in template to handle @CONFPATH@ correctly [#1711, #3329]
- clients/authconf.c: parse bracketed IPv6 addresses correctly [#3329, #3503]
- tests/test_authconf.c: add tests to parse IPv6 addresses (brackets and colons!) [#3329, #3503]
- tests/test_authconf.c: test that (comment) text after a section name is ignored [#3329]
- tests/test_authconf.c: auto-account expected_sections (to check parsed num_sections) [#3329]
- tests/NIT/nit.sh: generatecfg_nutauth(): only handle case WITH_SSL_CLIENT_CERTIDENT ... if WITH_SSL_CLIENT != none [#1711]

- tests/NIT/nit.sh: move check for usability of `certutil` or `openssl` into `prepare_NIT_certs()` where we would inevitably use them [#1711]
- tests/NIT/nit.sh: fix indentation [#1711]

2.175 2026-06-23 juslex <66561713+juslex@users.noreply.github.com>

- NEWS.adoc, docs/man/ragtech.txt, docs/nut.dict, drivers/ragtech.c: ragtech: fix CDC-ACM startup (raw mode \+ initial-poll retry), battery scaling, va override Startup robustness (issue #3500, Easy Pro 3200 VA GT on a Raspberry Pi) -----
----- The reporter's unit failed every boot with "no reply to initial status poll", even though it speaks the family-10 protocol correctly (its 30-byte main range decodes cleanly against the OEM Supervise 8 readings). The root cause, pinned down from an `strace` \+ the probe script, was the serial line discipline, not the protocol:
- A freshly enumerated CDC-ACM tty comes up in canonical mode (ICANON). In that mode the kernel buffers incoming bytes until it sees a newline. The firmware's replies are raw binary and never contain one, so the bytes were held in the kernel and the first poll always timed out. The driver had been relying on the port already being raw — which happened to be true on the author's host only because its `systemd` unit ran "`stty ... raw`" in an `ExecStartPre`, masking the bug. Without that external step (e.g. a plain "`upsdrcvtl start`"), the driver never worked. `upsdrcvtl` now sets raw 8N1 mode itself: `tcsetattr/clear ICANON` et al./`tcsetattr`, leaving the baud untouched (CDC-ACM ignores it at the wire), with `CLOCAL` set and `HUPCL` cleared so neither the reconfigure nor the eventual close toggles DTR, and DTR/RTS are forced low afterwards. The previous "NEVER call `tcsetattr`" comment was unfounded: NUT's own `ser_set_speed()` configures raw mode via `tcsetattr` and is used by ~50 serial drivers, the reporter observed no DTR pulse / shutdown, and this driver's shutdown path is a register write, not a DTR edge. * `upsdrcvtl` now retries the wake-up handshake and first poll a few times before calling `fatalx()`. Right after a cold boot the device may have only just enumerated and the firmware can need a moment to answer. Retrying lets the driver come up cleanly under the stock NUT `systemd` units (which start it as soon as the device node appears) instead of depending on a fatal exit plus a `Restart=on-failure` cycle to recover, which is noisy and only retries on a 5 s cadence. Combined with the raw mode above, no external `stty`, `sleep`, or `.device` binding is needed. * The firmware answers the handshake with a byte or two (e.g. `0xCA`). Some models are half-duplex and drop a command sent while that reply is still in flight; the handshake reply is now drained before the first poll (which only works because raw mode actually delivers those bytes). Other fixes ----- * Compute `battery.charge` as `raw / 2.55` so a full battery reports exactly 100 % instead of 100.2 % (drops the previous clamp). * Add a "va" `ups.conf` option to override the apparent-power rating. The model id byte is shared between product lines (a 3200 VA GT reports the same id as an Easy 2200 TI), so the driver cannot always pick the right VA from its table; the override corrects `ups.power.nominal`, `ups.realpower.nominal` and the computed `ups.load`. Verified: family-10 Easy 2000 TI on hardware (raw mode, handshake drain, battery 100.0, va=3200 override) and the 3200 VA GT by the reporter (full telemetry, matching Supervise 8). Driver bumped to 0.10, still `DRV_EXPERIMENTAL`.

2.176 2026-06-23 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsclient.c: `upscli_free_host_port_cert_data()`: only defined for SSL-capable builds [#3329]
- clients/upsclient.c: `upscli_free_host_port_cert_data()`: consider const-ness of freed strings [#3329]
- clients/upsclient.c: `upscli_free_host_port_cert()`: consider builds without SSL [#3329]
- drivers/dummy-ups.c: refactor with `upscli_authconf_update_conn_flags()` [#3329]
- clients/upsc.c, clients/upsclient.c, clients/upsclient.h, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsrw.c, clients/upsset.c, clients/upsstats.c, tools/nut-scanner/scan_nut.c: tools/nut-scanner/scan_nut.c, clients/*.c: refactor with `upscli_authconf_update_conn_flags()` [#3329]
- clients/upsc.c, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsrw.c, clients/upsset.c, clients/upsstats.c: clients/*.c: only consider `authconf` default `certverify` and `flags_ssl` if positive [#3329]
- tests/NIT/nit.sh: on MSYS2, copy (not symlink) root CA cert hash-named files right away [#1711]
- tests/NIT/nit.sh: when transplanting prepared certs to/from cache, fall back to dereferencing symlinks if `cp -prf` failed [#1711]
- tests/NIT/nit.sh: `TESTCERT_PATH_SEP`: there was too much of a good thing for WIN32 runs [#1711]

- tests/NIT/nit.sh: use TESTCERT_PATH_SEP more diligently (e.g. on WIN32 runs) [#3331, #1711]
- tests/NIT/nit.sh: die() with a timestamp too

2.177 2026-06-23 Boris Shor <boris@bshor.com>

- NEWS.adoc, data/driver.list.in, docs/nut.dict, drivers/idowell-hid.c: idowell-hid: claim GoldenMate LiFePO4 (06da:ffff) without hijacking other Phoenixtec devices GoldenMate 1000VA/800W LiFePO4 packs reuse the shared Phoenixtec vendor ID 0x06da:0xffff with the same -BMS- firmware and HID descriptor as the existing iDowell 0x075d:0x0300 device, so they fall back to the generic liebert-hid handler (bogus battery.runtime, etc). The 0x06da VID is shared: mge-hid claims it for AEG PROTECT NAS and liebert-hid is its default sink. Probe order is mge → idowell → liebert, so idowell can intercept GoldenMate before liebert provided it does not grab the other 0x06da devices. - Add a named PHOENIXTEC_VENDORID define and use it for the 0x06da:0xffff device-table entry (feeds the udev/nut-scanner parser). - Gate idowell_claim() on the device's -BMS-/Smart-Battery strings, like mge_claim() does for AEG, returning 0 for non-GoldenMate 0x06da so they fall through to liebert-hid / mge-hid. - Bump IDOWELL_HID_VERSION to 0.21 and add a NEWS.adoc entry. - List both GoldenMate variants in data/driver.list.in. Fixes: <https://github.com/networkupstools/nut/issues/3501>

2.178 2026-06-22 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, UPGRADING.adoc, clients/upsmon.c, clients/upssched.c, conf/upsmon.conf.sample.in, conf/upssched.conf.sample.in, docs/man/upsmon.conf.txt, docs/man/upsmon.txt, docs/man/upssched.conf.txt, docs/man/upssched.txt, docs/nut.dict: clients/upsmon.c, clients/upssched.c, docs: handle NOTIFYCMD and CMDSCRIPT values strictly as paths [#3499]
- docs/configure.txt: properly pass RPATH for linker not compiler in LDFLAGS [#3495]
- NEWS.adoc: update about recipe and code portability PRs [#3489, #3495, #3498]
- ci_build.sh: properly pass RPATH for linker not compiler in LDFLAGS [#3495]
- NEWS.adoc, clients/Makefile.am, clients/upsimage.c, clients/upsimagearg.h: clients/upsimage.c, clients/upsimagearg.h: adjust scaling for small (single-digit) ranges [#3469] Avoid lack of output when the range of a variable is small (e.g. 0-10) and division by zero occurred instead.
- NEWS.adoc, clients/upsc.c, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsmon.c, clients/upsrw.c, clients/upssched.c, clients/upsset.c, clients/upsstats.c: clients/*.c, NEWS.adoc: fflush() stderr and stdout before clean-up and exit, just in case [#3454]
- clients/upsclient.c: wherever we fail upscli_init*(), call upscli_cleanup() right away [#3454]
- clients/upsclient.c: pre-initialize SSL_CTX as NULL to be sure [#3454]

2.179 2026-06-21 Jim Klimov <jimklimov+nut@gmail.com>

- tests/.gitignore: GitIgnore tests/test_authconf if built [#3329]
- NEWS.adoc, docs/man/dummy-ups.txt, drivers/dummy-ups.c: dummy-ups: add NUT authconf support [#3329]

2.180 2026-06-21 Huang Rui <vowstar@gmail.com>

- NEWS.adoc, drivers/xppc-mib.c: feat(snmp-ups): map more xppc readings for Phoenixtec UPS cards - add battery.runtime, battery.voltage and input.frequency readings - add output.voltage.nominal and ups.firmware.aux readings - bump XPPC_MIB_VERSION to 0.41

2.181 2026-06-21 Doug Nazar <nazard@nazar.ca>

- server/upsd.c: AIX 64 uses a long for fd in pollfd
 - ci_build.sh: Tell compiler to pass -R option to linker instead of a remark
-

2.182 2026-06-20 penghong123 <2835950872@qq.com>

- data/driver.list.in: Add some new UPS models

2.183 2026-06-20 Doug Nazar <nazard@nazar.ca>

- include/timehead.h: Provide function declaration when needed
- drivers/powervar_cx_ser.c: AIX doesn't provide defines for those speeds.
- clients/upsc.c, clients/upscmd.c, clients/upslug.c, clients/upsrw.c: Ignore unused parameters
- Makefile.am: Switch to using -t to specify date as -d not always available
- configure.ac: Fix %zu format check
- configure.ac: Use lsgroup on AIX to check for valid groups
- m4/nut_compiler_family.m4: Check enable version of warnings to see if valid GCC will silently accept warning disable options, but then if there is an issue, warn about the option. "cc1: note: unrecognized command-line option *-Wno-unknown-warning-option* may have been intended to silence earlier diagnostics"
- scripts/python/module/Makefile.am: Don't try to copy permissions When the source files are on a different filesystem it's possible for the copy to fail. Seen on FreeBSD, when trying to copy from an NFS mount it would fail trying to copy the ACLs.
- tools/Makefile.am: Use full path for generated files When using relative paths Make will use VPATH to find the file in the source directory. If out of date, it will then try to update that file instead and fail on a read-only source directory.
- include/common.h, tools/nut-scanner/scan_eaton_serial.c: Add check for older IBM compilers. Newer compilers are based on Clang and errors on GCC.

2.184 2026-06-19 Jim Klimov <jimklimov+nut@gmail.com>

- tools/nut-scanner/scan_nut.c: follow up on NUT authconf support when scanning "old nut" servers [#3329]
- clients/upscclient.c, clients/upscclient.h: clients/upscclient.{c,h}: add pthreads support for first_host_cert-based list manipulation, and methods to remove items from list [#3329]

2.185 2026-06-17 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upscclient.h: update comments for methods about default_connect_timeout [#3329]
- tools/nut-scanner/scan_nut.c: nutscan_scan_ip_range_nut_authconf(): detect and load NUT auth conf file like other clients do [#3329]
- tools/nut-scanner/scan_nut.c: nutscan_scan_ip_range_nut_authconf(): detect desired timeout like other NUT clients do [#3329]

2.186 2026-06-16 Jim Klimov <jimklimov+nut@gmail.com>

- tools/nut-scanner/scan_nut.c: detect if authconf-related methods are present in libupscclient build we try to use [#3329]
 - docs/man/nutscan_scan_nut.txt, tools/nut-scanner/Makefile.am, tools/nut-scanner/nut-scan.h, tools/nut-scanner/scan_nut.c: tools/nut-scanner/*, docs/man/nutscan_scan_nut.txt: introduce nutscan_scan_ip_range_nut_authconf() and nutscan_scan_nut_authconf() with nutscan_nut_authconf_t argument [#3329] Make old methods wrappers of the new ones, so existing libnutscan clients continue working as they were (no breaking ABI change, just new methods).
 - scripts/obs/debian.rules, scripts/obs/nut.spec: update comments about SO_MAJOR_LIB* version hassle
 - docs/nut.dict: sort the dictionary after PR #3447
-

- NEWS.adoc: mention [PR #3448](#)
- NEWS.adoc: mention [PR #3456](#)
- docs/Makefile.am: fix ASPELL_ENV_LC_ALL for platforms that care
- NEWS.adoc: refer to [PR #3485](#)

2.187 2026-06-15 Jim Klimov <jimklimov+nut@gmail.com>

- docs/ci-farm-do-setup.adoc, docs/nut.dict: add notes about ZFS and 4KB disk sectors

2.188 2026-06-14 mbv06 <mbv06.dev@gmail.com>

- NEWS.adoc, data/driver.list.in, drivers/nutdrv_qx.c: nutdrv_qx: Fix armac subdriver failing on 7-byte interrupt reads Certain UPS models (e.g. SVC VP-1250-LCD) return a 7-byte report instead of the standard 6 bytes. This relaxes the strict equality check in the interrupt read loop and increases the requested read size to 8 bytes to safely accommodate these devices without dropping trailing bytes or timing out.

2.189 2026-06-12 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: update optional traceable toggles

2.190 2026-06-11 mbv06 <mbv06.dev@gmail.com>

- NEWS.adoc: docs: note Windows state pipe security update

2.191 2026-06-11 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/richcomm_usb.c, NEWS.adoc: bump DRIVER_VERSION and document the improvements [[#1768](#), [#3477](#)]

2.192 2026-06-10 dependabot[bot] <49699333+dependabot[bot]@users.noreply.github.com>

- .github/workflows/01-make-dist.yml: build(deps): bump bluwy/substitute-string-action from 3 to 4 Bumps [bluwy/substitute-string-action](#) from 3 to 4. - [Release notes](#) - [Commits](#) ---- updated-dependencies: - dependency-name: bluwy/substitute-string-action dependency-version: 4 dependency-type: direct:production update-type: version-update:semver-major ----
- .github/workflows/01-make-dist-worker.yml: build(deps): bump actions/upload-artifact from 4 to 7 Bumps [actions/upload-artifact](#) from 4 to 7. * [Release notes](#) * [Commits](#) ---- updated-dependencies: * dependency-name: actions/upload-artifact dependency-version: 7 dependency-type: direct:production update-type: version-update:semver-major ----

2.193 2026-06-09 mbv06 <mbv06.dev@gmail.com>

- docs/man/richcomm_usb.txt, drivers/richcomm_usb.c, tools/nut-scanner/scan_usb.c: Fix richcomm USB matching options
- drivers/richcomm_usb.c: drivers: preserve libusb context across USB reopens (richcomm only)

2.194 2026-06-08 mbv06 <mbv06.dev@gmail.com>

- drivers/dstate.c: drivers: fix Windows state pipe permissions
-

2.195 2026-06-08 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.h, clients/nutclient.cpp, clients/nutclient.h, scripts/perl/UPS/Nut.pm, scripts/python/module/PyNUT.py.in: clients/authconf.h, clients/nutclient.{h,cpp}, scripts/perl/UPS/Nut.pm, scripts/python/module/PyNUT.py.in: revise ability to parse nutauth.conf files [#3329]
- scripts/perl/UPS/Nut.pm: fix old trailing whitespaces

2.196 2026-06-07 Jim Klimov <jimklimov+nut@gmail.com>

- include/nutconf.hpp, common/nutconf.cpp: add ability to parse nutauth.conf files [#3329]
- clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.h, scripts/perl/UPS/Nut.pm, scripts/python/module/PyNUT.py.in: clients/nutclient*.{h,cpp}, scripts/perl/UPS/Nut.pm, scripts/python/module/PyNUT.py.in: add ability to parse nutauth.conf files [#3329]

2.197 2026-06-06 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsclient.c: upscli_authenticate(): reduce buf sizes for result to fit UPSCLI_NETBUF_LEN [#3329]
- clients/upsimage.c, clients/upsset.c, clients/upsstats.c: clients/ups*.c: comment in CGI clients about auto-login (not wise for that use case) [#3329, #3411]
- clients/upsc.c: support upscli_authenticate_authconf() [#3411, #3329]
- clients/upsclient.c, clients/upsclient.h, clients/upscmd.c, clients/upsmon.c, clients/upsrw.c, docs/man/Makefile.am, docs/man/upsc.txt, docs/man/upscli_authenticate.txt, docs/man/upscmd.txt, docs/man/upslog.txt, docs/man/upsrw.txt: clients/upsclient.{c,h}, docs/man, upscmd.c, upsrw.c, upsmon.c: refactor with upscli_authenticate() to unite different tricks [#3329] Introduce upscli_authenticate for completeness.
- docs/Makefile.am: fix SOURCE_CHECK_UNICODE for out-of-tree builds

2.198 2026-06-05 Jim Klimov <jimklimov+nut@gmail.com>

- clients/Makefile.am, configure.ac: configure.ac: revise LIBSSL_LIBS vs "-pthread" option (from NSS occasionally) for C++ builds [#3453] Address error: argument unused during compilation: *-pthread*
- Makefile.am, SECURITY.md, docs/Makefile.am: SECURITY.md: introduce a policy that can be seen on GitHub Describe the current situation with support of NUT versions and the parties involved, and how to responsibly disclose vulnerabilities.

2.199 2026-06-03 Jim Klimov <jimklimov+nut@gmail.com>

- .github/workflows/01-make-dist-worker.yml, .github/workflows/01-make-dist.yml, .github/workflows/05-codeql.yml, .github/workflows/PyNUTClient.yml: .github/workflows/*.yml: bump actions versions Originally proposed by dependabot, but we cannot put its Markdown commit comments into ChangeLog.pdf, so reposting.
- .github/dependabot.yml: introduce scanning for obsolete GHA versions
- .github/workflows/01-make-dist.yml: update comments about the Checks message [#1400]
- .github/workflows/01-make-dist.yml: when dispatching the workflow for a branch iteration, use its name (ref) not commit [#1400]
- .github/workflows/01-make-dist.yml: delete old artifacts OTHER than the one just created [#1400] Before the recent refactoring, the logic was to first build the tarball, then if it succeeded - clean up the storage from obsolete ones, and quickly publish the new one. Now that a separate workflow does the uploads, we can at best ensure deletion of obsolete tarballs after hopefully having the new one up there (had enough space and all that). Also take care to not delete the newly made artifact ;)

- NEWS.adoc: update for FreeBSD and Jenkins bumps
- NEWS.adoc: Rearranged the GHA job to prepare Dist and Docs Tarballs [#1400, #3460]
- .github/workflows/01-make-dist.yml: launch "GHA-01: Tarballs Build Worker" definition from the target branch (if a PR) [#1400]
- configure.ac: pass -Wno-unused-command-line-argument also to CLANG LDFLAGS
- .github/workflows/01-make-dist.yml: DRY: define artifact_name once [#1400]

2.200 2026-06-02 Jim Klimov <jimklimov+nut@gmail.com>

- .github/workflows/01-make-dist-worker.yml, .github/workflows/01-make-dist.yml: .github/workflows/01-make-dist{,-worker}.yml: retell the story of two workflows [#1400]
- .github/workflows/01-make-dist.yml: reduce the permissions needed for artifact manipulation [#1400]

2.201 2026-06-01 Jim Klimov <jimklimov+nut@gmail.com>

- Jenkinsfile-dynamatrix: use new countStagesPerNode feature from the library in notifyHandlerAlert
- Jenkinsfile-dynamatrix: introduce dynacfgPipeline.notifyHandlerAlert (to report looping RESTART dynamatrix cells)

2.202 2026-05-31 Jim Klimov <jimklimov+nut@gmail.com>

- configure.ac: ignore recent clang warning about unused-command-line-argument

2.203 2026-05-31 Doug Nazar <nazard@nazar.ca>

- common/common.c, drivers/upsdrvquery.c, include/common.h, server/sstate.c: Use non-blocking connect() for AF_UNIX sockets to ensure we don't stall. If a driver has become unresponsive, connect() will block after the listen() backlog has been exhausted, causing the main upsd to also become unresponsive.

2.204 2026-05-28 Jim Klimov <jimklimov+nut@gmail.com>

- tools/check-source-nonascii.pl: fix syntax for hash length
- ci_build.sh: fix normalize_path_perl() syntax for hash length; avoid newlines

2.205 2026-05-27 Jim Klimov <jimklimov+nut@gmail.com>

- docs/config-prereqs.txt: document JDK21+ as the requirement for NUT CI build farm agents since spring 2026
 - ci_build.sh: normalize_path() before reporting it at start of actual work about BUILD_TYPE
 - ci_build.sh: configure_nut(): report the PATH value ultimately used (maybe normalized)
 - ci_build.sh: propose shell vs perl (faster?) implementations of normalize_path()
 - drivers/bicker_ser.c, drivers/nut-libfreeipmi.c, drivers/phoenixcontact_modbus.c, drivers/socomec_jbus.c, drivers/usbhid-ups.c: drivers/*.c: replace Unicode in comments with ASCII equivalents
 - drivers/socomec_jbus.c: clean up trailing whitespace
 - docs/documentation.txt, docs/man/nut.txt: replace Unicode long dashes with acsii --
-

- drivers/libwinhid.h: replace Unicode long dashes with `acsii --` and arrow with `=>`
- docs/developers.txt, docs/nut.dict: docs/developers.txt: add a chapter on ASCII vs Unicode
- Makefile.am, configure.ac, docs/Makefile.am, tools/Makefile.am, tools/check-source-nonascii.pl: configure.ac, Makefile.am, docs/Makefile.am, tools/check-source-nonascii.pl: add unicode-in-sources detection to spellcheck On some platforms, at least the `file` type parsing fails for text with some but not other wide characters (so it is "data" and not some sort of "text"). This test was added for C sources as well, but keep in mind that the final authority is the actually used compilers - which are free to complain or not.
- .github/pull_request_template.md: update with a section on Unicode characters
- configure.ac: report the tested compiler location just before `AC_PROG_CC` etc. which might fail
- configure.ac: `unset CCACHE_DIR` if provided but empty Notably, `ccache-3.7.12` in FreeBSD 15 breaks CI build with that complaint.
- drivers/ragtech.c: pre-declare variables for portability [#3447]
- drivers/ragtech.c: replace Unicode long dashes with `acsii --` [#3447]
- docs/man/ragtech.txt: replace Unicode long dashes with `acsii --` [#3447]

2.206 2026-05-26 Jim Klimov <jimklimov+nut@gmail.com>

- docs/config-prereqs.txt, docs/nut.dict: revise with FreeBSD 15 lessons learned
- configure.ac: if developer enables configure debug, only wipe `DOCTESTDIR` after confirming we can build docs we want
- drivers/ragtech.c: fix compiler warnings about `%u` vs. conditionally picked numbers [#3447] ... which are signed int by default

2.207 2026-05-25 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, UPGRADING.adoc, docs/man/upsd.conf.txt, server/netssl.c: server/netssl.c: `ssl_init()`: abort with `fatalx()` and better explanations when SSL setup was requested but failed [#3331]
- tests/NIT/nit.sh: `check_NIT_certs_NSS()`: fix the check for `*.txt` and `*.db` file patterns [#3331, #1711]
- tests/NIT/nit.sh: wrap long `openssl` lines; fix a missed validity-time spec [#1711]
- tests/NIT/nit.sh: `check_NIT_certs_NSS()`: verify old run-time toolkit vs. newer (cached) DB files [#3331, #1711] Auto-repair the trusted NSS DB set(s) of files if we can. Primarily relevant for build agents where same cache is used by NSS toolkits a decade apart.
- docs/nut.dict: add opcodes
- configure.ac: add `-Wno-thread-safety-negative` to clang flags (seen on FreeBSD builds)

2.208 2026-05-24 Pedro Cunha <pedroagracio+nut@gmail.com>

- drivers/libusb1.c: use explicit context; drop per-close `libusb_exit()` [#598] The original code called `libusb_init(NULL)` on every `nut_libusb_open()` and `libusb_exit(NULL)` on every `nut_libusb_close()`, bumping and decrementing the default context's refcount. That violates the documented contract for `libusb_exit`, which "should be called after closing all open devices and before your application terminates", and on certain firmware it wedges indefinitely: when `libusb_exit` hits the `1→0` transition it tears the context down, which waits on `libusb`'s internal sync primitives for outstanding URBs to drain. URBs orphaned by `libusb_reset_device` or by an unexpected device disconnect never drain, so the wait never returns. The deadlock is reachable from any reconnect path. It was first recognized by 4f84b7f92 ("don't `libusb_exit()` when closing a previously opened device"), which dropped the call in `nut_libusb_open()`'s rematch loop. The fallthrough reconnect path in `qx_command` and the rest of the driver lifecycle still hit it. With the companion change in `nutdrv_qx.c` (a4813f868) that escalates persistent

LIBUSB_ERROR_OVERFLOW to libusb_reset_device, the deadlock window opens on every escalation and was hit reproducibly on the 0665:5161 Cypress USB-serial bridge family (Salicru SPS, Ippon, ViewPower, Voltronic Power UPSes; see [#598](#), [#993](#), [#2453](#)). Switch to an explicit libusb context owned end-to-end: - Initialize once on the first nut_libusb_open() and register a matching one-shot atexit handler. - Remove libusb_exit() from nut_libusb_close(); the close path now only calls libusb_close(), which is non-blocking. - Replace libusb_get_device_list(NULL, ...) with the explicit context so the default context is never touched.

- Bound the atexit libusb_exit() with a SIGALRM-based 2s timeout (POSIX) so a deadlock-prone teardown at shutdown can't wedge supervisor stop sequences (e.g. systemctl stop). Sub-millisecond on a healthy context. Bump USB_DRIVER_VERSION from 0.53 to 0.54. Validated on a Salicru SPS 1500 ONE BL (0665:5161) by authorize-toggle stress reproduction (8 cycles, clean reconnect each time, same driver PID throughout) and by a natural OVERFLOW event captured in soak (counter-gated logic at 1/3, next poll succeeded, no escalation; driver PID stable for 22h\+).

2.209 2026-05-22 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c, clients/authconf.h, docs/man/upscli_dump_authconf_item.txt: clients/authconf.{c,h}, docs/man/upscli_dump_authconf_item.txt do not specify "FILE *restrict stream", some platforms think "restrict" is the variable name [[#3329](#)]
- lib/libupsclient-config.in: this should consider LIBSSL_LDFLAGS_RPATH [[#3331](#)]
- lib/libnutclient.pc.in: this now Requires: @LIBSSL_REQUIRES@ [[#3331](#)]

2.210 2026-05-21 Jim Klimov <jimklimov+nut@gmail.com>

- tests/Makefile.am: test_authconf: consider LIBSSL flags [[#3329](#), [#1711](#)] Builds with NSS tend to fail on some platforms due to not locating libnss3.so (which is hidden from common searches in an nss/mps subdirectory)

2.211 2026-05-20 Pedro Cunha <pedroagracio+nut@gmail.com>

- drivers/nutdrv_qx.c: qx_command(): reset USB device after persistent LIBUSB_ERROR_OVERFLOW [[#598](#)] LIBUSB_ERROR_OVERFLOW has been handled as a benign, retry-on-next-poll condition (grouped with LIBUSB_ERROR_TIMEOUT and the default case) ever since the original blazer import. For most devices a one-off oversized interrupt-IN frame is indeed transient. But some Cypress USB-serial bridge firmware (VID:PID 0665:5161; Salicru SPS, Ippon, ViewPower and various Voltronic Power UPSes) wedges the endpoint once it overruns: every subsequent interrupt read returns OVERFLOW and the driver spins in the stale-data loop until an external USB-level reset. This is the chronic hang reported in [issue #598](#). Distinguish the two cases with a small consecutive-overflow counter: the first QX_USB_OVERFLOW_RESET_TRIES-1 overflows are retried on the next poll (so genuine transients cost nothing but a skipped cycle), and a sustained run escalates to usb_reset() \+ reconnect, reusing the recovery path already used for PIPE/ETIME/IO errors. Any clean read resets the counter. Captured on a Salicru SPS 1500 ONE BL: an isolated overflow recovers on the next poll, while a sustained lockup (dozens of consecutive overflows within seconds) is cleared only by the device reset, matching the behaviour of an external usb_resetter --reset-device.

2.212 2026-05-20 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsstats.c: ups_connect(): avoid potential NULL dereference in debug/failure printouts

2.213 2026-05-19 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, clients/upsc.c, clients/upscmd.c, clients/upsimage.c, clients/upslog.c, clients/upsrw.c, clients/upsset.c, clients/upsstats.c, docs/man/upsc.txt, docs/man/upscmd.txt, docs/man/upsimage.cgi.txt, docs/man/upslog.txt, docs/man/upsrw.txt, docs/man/upsset.cgi.txt, docs/man/upsset.conf.txt, docs/man/upsstats.cgi.txt: clients/{upscmd,upsrw,upslog,upsstats,upsset,upsimage}.c, docs/man/*.txt: add support for nutauth.conf [[#3329](#)]
- clients/upsclient.c: upscli_cleanup(): call upscli_free_authconf_list() [[#3329](#)]

- clients/upsclient.c, docs/man/upscli_add_host_cert.txt: refactor upscli_find_host_port_cert() with a "verbose" option, and extend upscli_add_host_port_cert() with check for existing entries [#3329]
- clients/authconf.c, drivers/libusb0.c, include/Makefile.am, include/strcasestr-static.h: include/strcasestr-static.h, clients/authconf.c, etc.: promote code from drivers/libusb0.c to be a bit more shared [#3329] Follows up from commits b91e34edc79 and f951dce8deb
- clients/upsclient.c: upscli_find_host_cert(), upscli_add_host_cert(): avoid "s_host" varname which confuses WIN32, illumos and other builds [#3329]
- clients/authconf.c, docs/man/nutauth.conf.txt, tests/NIT/nit.sh: set_authconf_val(): support USERNAME as alias of USER [#3329] Follow the NUT Networked Protocol keywords a bit more closely.
- tests/NIT/nit.sh: do not export NUT_AUTHCONF_FILE to NIT.env [#3329, #1711] Depending on context, whether "none" or a real path may be useful or toxic to custom developer test works.
- clients/upsclient.c: AuthCertificate(), upscli_sslinit(): refactor NSS server certname validation [#3331] It is NOT about just host name (URL) matching per our spec.
- clients/upsclient.c: revise logging from AuthCertificate*() methods [#3329]
- tests/NIT/nit.sh: generatecfg_upsmon_add_SSL(): consider CERTHOST with non-default NUT_PORT now [#3329]
- conf/upsmon.conf.sample.in, docs/man/upsmon.conf.txt: clarify CERTHOST with non-default NUT_PORT [#3329]
- clients/upsclient.c: upscli_add_host_cert(): fix reporting for plain legacy "hostname" input [#3329]
- clients/upsclient.c: BadCertHandler(): cast printing of port [#3329]
- clients/upsclient.c: upscli_sslinit(): wrap long lines in NSS part [#3329]
- tests/NIT/Makefile.am: check-NIT-devel: minimize nut_version.h regeneration attempts [#1711]
- tests/NIT/nit.sh: fix passing server cert trust to NSS DB [#3331, #1711]
- tests/NIT/nit.sh: fix passing server cert usage to NSS DB [#3331, #1711]
- tests/NIT/nit.sh: fix passing root CA cert trust to NSS DB [#3331, #1711]

2.214 2026-05-18 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/apc_common.c, drivers/apc_common.h: add Axel Gembe copyright to apc_common as the code is at the very least inspired by the original code from apc-modbus

2.215 2026-05-18 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsclient.c: upscli_find_host_port_cert(): debug-trace whether this succeeded [#3329]
- clients/upsclient.c: upscli_add_host_cert(): revise splitting apart the "hostname" which may come from authconf section name [#3329]
- clients/upsclient.c: upscli_init2(): quiesce reaction to NUT_QUIET_INIT_SSL=false [#1711]
- clients/upsclient.c: upscli_init_authconf(): if the "ac" refers to a CERTHOST, call upscli_add_host_cert() on it [#3329]
- clients/upsclient.c: upscli_init_authconf(): debug-trace the authconf pointer [#3329, #1711]
- clients/authconf.c, clients/authconf.h, docs/man/upscli_dump_authconf_item.txt, tests/test_authconf.c: clients/authconf.{c,h}, docs/man/upscli_dump_authconf_item.txt, tests/test_authconf.c: extend upscli_dump_authconf_{list,item}() with an option to show/hide passwords [#3329]
- NEWS.adoc, UPGRADING.adoc, clients/upsc.c, docs/man/upsc.txt: clients/upsc.c, docs: add support for "-A /path/to/nutauth.conf" [#3329, #3411]

- clients/upsclient.c: document better the upscli_init*() and upscli_sslinit() methods [#3329]
- clients/authconf.c: handle_authconf_args(): revise logging of parsed lines, especially malformed and sensitive ones [#3329]
- tests/test_authconf.c: consult presence of NUT_AUTHCONF_FILE envvar to parse and print out that file [#3329, #1711]
- tests/NIT/nit.sh: add support for nutauth.conf file testing, add a read-only user [#3329, #1711, #3411] Call new generatecfg_nutauth() after we do generatecfg_upsdusers_trivial() (and "coincidentally" in one case generatecfg_upsd_add_SSL() as well).
- clients/upsclient.c: upscli_add_host_cert(), upscli_find_host_cert(): refactor with shared get_port_from_string() logic [#3329]
- drivers/apcmicrolink.c: drop pragma to ignore a warning unknown to old clang [#3406]

2.216 2026-05-17 juslex <66561713+juslex@users.noreply.github.com>

- NEWS.adoc, data/driver.list.in, docs/man/Makefile.am, docs/man/ragtech.txt, docs/nut.dict, drivers/Makefile.am, drivers/ragtech.c, include/wincompat.h: ragtech: new driver for Ragtech UPSes (USB CDC-ACM, family 10) Targets Brazilian-built Ragtech "Easy Pro" / NEP / TORO / INNERGIE / OneUP devices that present themselves as USB CDC-ACM (VID 0x04D8, PID 0x000A, Microchip PIC firmware). Validated end-to-end against an Easy 2000 TI (reg 0x9A model id = 16) read out of a working OneUP Nitro 2000. Protocol — three opcodes observed in OEM traffic: 0x01 ADDR_HI ADDR_LO VALUE CKSUM write byte 0x02 ADDR_HI ADDR_LO MASK CKSUM AND mask (atomic bit-clear) 0x04 ADDR_HI ADDR_LO COUNT CKSUM read range CKSUM = (ADDR_HI \+ ADDR_LO + VALUE) & 0xFF The CDC-ACM channel ignores baud at the wire but DTR/RTS are interpreted by some Ragtech families as a remote shutdown signal — the driver forces both low after open and does NOT call ser_set_speed() to avoid the tcsetattr() DTR pulse that some Linux tty drivers perform. The full 30-byte main range (0x80..0x9D) plus V_IOUTCALIB (0xF3) and the oscillator calibration pair (0x202/0x203) are read; output frequency is interpolated per devices.xml formula rather than hard-coded. Twenty models from the family-10 device table populate ups.model, ups.realpower.nominal, output.voltage scaling, output.current scaling and battery voltage scaling. input.voltage.nominal switches between 115V and 220V dynamically based on the measured input. Instcmds implemented: shutdown.stayoff — aa 02 00 80 fe (clear AUTOSTART) \+ aa 01 00 98
 A. Byte-for-byte identical to the OEM supsvc sequence. Validated: cuts output and stays off until manual power-on, in both AC and battery modes. shutdown.return — falls back to shutdown.stayoff with a warning. The atomic OR opcode that would set F_AUTOSTART has not been captured yet; read-modify-write with 0x01 sends the bytes but does not trip the firmware's shutdown state machine. shutdown.stop — aa 01 00 98 00 (write V_SHUTDOWNTIMER = 0). Confirmed to abort an armed countdown. test.battery.start.deep — bit writes on regs 0x90 / 0x95 (fullDischarge in devices.xml). Flagged as unverified for the same reason as shutdown.return. Read primitive uses ser_get_buf() in a manual loop rather than ser_get_buf_len() (which discards partial reads on timeout — CDC fragments replies into 1-8 byte chunks). ser_flush_io() is used before each TX to clear the kernel CDC buffer (ser_flush_in() only does a select\+read loop that misses queued bytes under O_NONBLOCK). Reverse engineering primarily from the OEM devices.xml table for register layout / scaling / flags / actions, and from strace of the OEM supsvc binary capturing both read polls and the LED/shutdown write sequences. Cross-checked against UPS_ESP32_tinySrv (https://github.com/antunesls/UPS_ESP32_tinySrv).

2.217 2026-05-15 Lukas Schmid <lukas.schmid@netcube.li>

- docs/man/apcmicrolink.txt, docs/nut.dict, drivers/apcmicrolink.c: update docs and spellcheck dictionary

2.218 2026-05-15 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c, docs/man/upscli_read_authconf_file.txt: add support for NUT_AUTHCONF_FILE and/or NUT_AUTHCONF_PATH envvars to locate exactly one nutauth.conf candidate or bail out [#3329]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: add rudimentary support for SSLConfig_CERTHOST with a port number [#3329]
- clients/authconf.c, clients/upsclient.c, clients/upsclient.h, docs/man/Makefile.am, docs/man/upscli_add_host_cert.txt: clients/upsclient.{c,h}, clients/authconf.c, docs: introduce upscli_add_host_port_cert() et al [#3329]

- clients/authconf.c, tests/test_authconf.c: handle_authconf_args(): when we finish parsing a section, upscli_add_host_cert() if applicable [#3329]
- clients/authconf.c, clients/authconf.h, docs/man/nutauth.conf.txt, docs/man/upscli_create_authconf_item.txt: clients/authconf.{c,h}, docs/man/upscli_create_authconf_item.txt, docs/man/nutauth.conf.txt: add CERTHOST field support [#3329]

2.219 2026-05-14 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/Makefile.am: add missing apc_common.h to distributed headers
- drivers/apcmicrolink-maps.c, drivers/apcmicrolink-maps.h, drivers/apcmicrolink.c: extract descriptor helpers and fix map includes

2.220 2026-05-14 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c: upscli_get_authconf_item(): add ifdef-ed away very optional debug messages [#3329]
- tests/test_authconf.c: print TAP-style progress tracking| add tests for upscli_get_authconf_item() modification of the list [#3329]
- clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/upscli_find_authconf_item.txt, tests/test_authconf.c: clients/authconf.{c,h}: introduce upscli_get_authconf_item() for auto-merged items [#3329]
- docs/man/Makefile.am: fix after reshuffle nutauth.conf related methods vs population of man pages [#3329]
- clients/authconf.c, docs/man/upscli_create_authconf_item.txt: revise upscli_merge_authconf_item() function vs. cloning of empty strings [#3329]
- clients/authconf.c, docs/man/upscli_create_authconf_item.txt: revise upscli_clone_authconf_item() and upscli_merge_authconf_item() functions vs. cloning of USER name [#3329]
- clients/authconf.c: refactor with separation of upscli_add_authconf_item(name) and upscli_add_authconf(node) [#3329]
- clients/authconf.c: upscli_find_authconf_item(): refine return of global_defaults and zearch in empty list [#3329]

2.221 2026-05-13 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/apcmicrolink.c, drivers/apcmicrolink.h: fix switch warnings and replace flag bit-fields

2.222 2026-05-13 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/upscli_create_authconf_item.txt, docs/man/upscli_get_authconf_item.txt, docs/man/upscli_find_authconf_item.txt: clients/authconf.{c,h}, docs: introduce upscli_clone_authconf_item() and upscli_merge_authconf_item() functions [#3329]
- clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/nutauth.conf.txt, ..._authconf.txt ⇒ upscli_create_authconf_item.txt, ..._authconf.txt ⇒ upscli_dump_authconf_item.txt, ..._authconf.txt ⇒ upscli_find_authconf_item.txt, docs/man/upscli_free_authconf_list.txt, docs/man/upscli_get_authconf_list.txt, docs/man/upscli_init.txt, ..._authconf.txt ⇒ upscli_read_authconf_file.txt, tests/test_authconf.c, clients/authconf.{c,h}, tests/test_authconf.c, docs/man/auth, docs/man/Makefile.am: rename file/list/item/section methods consistently with their purpose [#3329]
- clients/authconf.c: upscli_create_authconf(): handle failed calloc() by returning NULL; fail fatally in upscli_add_authconf() if needed [#3329]
- docs/man/Makefile.am, docs/man/upscli_create_authconf.txt, ..._authconf_list.txt ⇒ upscli_dump_authconf.txt, docs/man/upscli_find_authconf.txt, docs/man/upscli_free_authconf_list.txt, docs/man/upscli_get_authconf_list.txt: docs/man/auth.txt, docs/man/Makefile.am: reshuffle nutauth.conf related methods vs population of man pages [#3329]

- clients/authconf.c, docs/man/nutauth.conf.txt: revise section title parsing, INCLUDE handling, naming of [_global_defaults], code comments etc. [#3329]
- clients/authconf.c, clients/authconf.h: clients/authconf.{c,h}: refactor static upscli_add_authconf() with a public upscli_create_authconf() [#3329]
- clients/authconf.c: fix static analysis warnings [#3329]
- tests/test_authconf.c: add a Non-exact match test case [#3329]
- clients/authconf.c: refactor upscli_find_authconf() with upscli_normalize_auth_section_parts() [#3329]
- NEWS.adoc, drivers/nutdrv_qx_megatec.c: drivers/nutdrv_qx_megatec.c: add QX_FLAG_ABSENT to "ups.firmware" so devices with all-spaces in that field can still be recognized [#3436] Co-authored-by: Jim Klimov <jimklimov+nut@gmail.com> Co-authored-by: Siravijbb <siravijbb@gmail.com>

2.223 2026-05-12 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/apc_modbus.c: remove unused function

2.224 2026-05-12 Jim Klimov <jimklimov+nut@gmail.com>

- clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/upscli_find_authconf.txt: clients/authconf.{c,h}, docs: further refactor with upscli_normalize_auth_section_parts() which we can share and expose [#3329]
- clients/authconf.c: handle_authconf_args(): revise section title line parsing [#3329]
- clients/authconf.c: upscli_splitauth(): reject empty non-NULL sect_name [#3329]
- clients/authconf.c: upscli_splitauth(): if "port" is a non-numeric string, try to resolve it in the system naming database [#3329]
- scripts/obs/debian.nut-client.install, scripts/obs/debian.nut-client.manpages, scripts/obs/nut.spec: know about nutauth.conf samples [#3329]
- conf/nutauth.conf.sample.in, configure.ac, conf/Makefile.am, conf/.gitignore: add example config for nutauth.conf [#3329]
- tests/test_authconf.c: add a dev-test for discovery of user/site nutauth configs [#3329]
- tests/test_authconf.c: honour NUT_DEBUG_LEVEL [#3329]
- clients/authconf.c, clients/authconf.h, docs/man/upscli_read_authconf.txt: clients/authconf.{c,h}, docs/man/upscli_read_authconf.txt: add searching for default nutauth.conf in user home or NUT_CONFPATH locations [#3329]
- clients/upsclient.c, clients/upsclient.h, docs/man/Makefile.am, docs/man/upscli_init.txt: clients/upsclient.{c,h}, docs/man/upscli_init.txt: introduce upscli_init_authconf() [#3329]
- clients/authconf.c: parse_authconf_file(): call check_perms(filename) [#3329, #3359]
- common/common.c, include/common.h, server/upsd.c, server/upsd.h, server/user.h: common/common.c et al: relocate check_perms() from upsd to common code base [#3359]
- clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/upscli_find_authconf.txt, docs/nut.dict: clients/authconf.{c,h}, docs/man/upscli_find_authconf.txt: refactor with upscli_split_auth_section() [#3329]
- clients/authconf.c, clients/authconf.h, docs/man/upscli_dump_authconf_list.txt, tests/test_authconf.c: clients/authconf.{c,h}, tests, docs: extend upscli_dump_authconf{,_list} methods with "for_debug" option [#3329]
- tests/test_authconf.c: drop "Mocks for functions usually provided by libcommon" [#3329]
- common/common.c: cosmetic fix
- docs/Makefile.am: add ASPELL_NUT_COMMON_ARGS to preclude mangling of nut.dict on some platforms when checking interactively

2.225 2026-05-12 Doug Nazar <nazard@nazar.ca>

- common/common.c, drivers/main.c, include/common.h: drivers/main.c: Split backgrounding driver into two stages This is needed in particular with libusb on linux as it will create a thread to handle either netlink or udev messages. If this is created before we background, the driver will hang during exit trying to join the non-existent thread in the new process. Shows this warning while stopping the driver. Stopping /run/nut/usbhid-ups-xxxx.pid failed, retrying harder: Success

2.226 2026-05-11 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/apcmicrolink.c: fix commands and writes sometimes not working and allow configuration of command source
- drivers/Makefile.am, drivers/apc_common.c, drivers/apc_common.h, drivers/apc_modbus.c, drivers/apc_modbus.h, drivers/apcmicrolink.c, drivers/apcmicrolink-maps.h, drivers/apcmicrolink-maps.c: Variable cleanup and commons creation between apc-modbus and microlink

2.227 2026-05-11 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc, clients/Makefile.am, clients/authconf.c, clients/authconf.h, docs/man/Makefile.am, docs/man/nutauth.conf.txt, docs/man/upscli_dump_authconf_list.txt, docs/man/upscli_find_authconf.txt, docs/man/upscli_get_authconf_list.txt, docs/man/upscli_read_authconf.txt, docs/nut.dict, tests/Makefile.am, tests/test_authconf.c: Introduce NUT "authconf" file support [#3329] First PoC from AI, slightly modified in review, following the spec requested in the GitHub issue. Stepping stone for further work.
- server/upsd.c: update_sysmaxconn(): default the sysmaxconn value to "MAX - RESERVE_FD_COUNT_UPSD" on platforms where it might matter [#3365]
- server/upsd.c: update_sysmaxconn(): check that detected sysmaxconn_hard is big enough to be useful considering the RESERVE_FD_COUNT_UPSD set aside AND a few actual connections [#3365]
- server/upsd.c: update_sysmaxconn(): revise warning message about hard limit [#3365]
- server/upsd.c: be sure to re-detect sysmaxconn after (re-)loading config [#3365]
- NEWS.adoc, UPGRADING.adoc, docs/nut.dict, server/upsd.c, server/upsd.h: server/upsd.{c,h}, NEWS.adoc, UPGRADING.adoc: consider getrlimit() permissions vs. MAXCONN request [#3365]
- NEWS.adoc, server/conf.c, server/upsd.c, server/upsd.h: server/conf.c, server/upsd.{c,h}, NEWS.adoc: if we hit "Too many open files" during configuration reload, close oldest client connection and retry [#3365]
- server/upsd.h: update © heading and whitespace styling
- docs/nut.dict: update for dllldd [#3420]
- NEWS.adoc: update about scripts/Windows/dllldd.sh helper script refinements [#3420]
- docs/man/upscli_add_host_cert.txt: update that this call works also with OpenSSL (since NUT v2.8.6) [#3331]
- docs/download.txt, docs/nut.dict: Upload NUT-for-Windows-x86_64-RELEASE-2.8.5-1-fixNSS.7z with NSS dependency libraries missing in original NUT v2.8.5 release [#3420]
- appveyor.yml: allow ln to create hard links, and tell 7-zip to store them as such, not as independent files (subject to 7zip format du jour)
- scripts/Windows/dllldd.sh: Revise use of TEMPFILE_REC to avoid grep if it is empty [#3420]
- Makefile.am: install-win-bundle-thirdparty: revise progress message markup for easier troubleshooting [#3420]
- scripts/Windows/dllldd.sh: dllldd_with_strings(): filter away OUT_TOOLS hits if known [#3420]
- scripts/Windows/dllldd.sh: dllldd_with_strings(): quick bail-out with TEMPFILE_REC [#3420]

- scripts/Windows/dllldd.sh: revise cleanup of TEMPFILE_REC; create it earlier in dlllddir() [#3420]
- scripts/Windows/dllldd.sh: dllldd(): use TEMPFILE_REC if defined by caller [#3420]
- scripts/Windows/dllldd.sh: dllldd_with_tools(), dllldd(): optimize looped iteration of search paths we prepared [#3420] Avoid calls to `tr ':\n'` that can be done once.
- scripts/Windows/dllldd.sh: relocate TEMPFILE_REC declaration to before first use-mention [#3420]
- scripts/Windows/dllldd.sh: filter_away_system_DLLs(): add some more OS-provided common hits to exclusion list [#3420]
- scripts/Windows/dllldd.sh: filter_away_system_DLLs(): add DWrite.dll [#3420] Some OS releases have Microsoft DirectWrite for high-quality text rendering, anti-aliasing, and font management. . . not something we really care about in NUT, and part of OS footprint we do not drill into.
- scripts/Windows/dllldd.sh: dllldd_with_tools(), dllldd(): optimize SEARCH_INPUT_PATH discovery [#3420] Instead of a loop with `dirname` calls (spawning those processes takes a few seconds in MSYS2 on Windows), use one `sed` process.
- scripts/Windows/dllldd.sh: dllldd_with_strings(): ignore also "lib%.dll" hits [#3420]
- scripts/Windows/dllldd.sh: accept also outputs from "objdump" variants that return "NEEDED libname" [#3420]
- scripts/Windows/dllldd.sh: when given a CLI argument, check if it is a directory (to run dlllddir) before diving in with dllldrec [#3420]
- scripts/Windows/dllldd.sh: accept also outputs from "ldd" variants that only return path (no parenthesis with hex) [#3420]
- scripts/Windows/dllldd.sh: allow to parameterize DLLEXT_REGEX [#3420] At least, allow to test/iterate the script with non-Windows builds too.
- scripts/Windows/dllldd.sh: dllldd_with_strings(): filter away some bogus strings outputs and findings [#3420]
- scripts/Windows/dllldd.sh: more filter_away_system_DLLs() [#3420]

2.228 2026-05-10 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/Windows/dllldd.sh: dllldd_with_tools(): define and clean-up TEMPFILE_REC in the outer-most call if we stack them [#3420] This should help directory crawlers etc. ignore paths already walked by some previous recursion in this scripted execution.
- scripts/Windows/dllldd.sh: dllldd_with_tools(): do not bail out if we found some but not all DLLs via objdump; feed only the names we did not resolve to ldd [#3420]
- scripts/Windows/dllldd.sh: consider also SEARCH_INPUT_PATH based on where the input EXE/DLL files are [#3420] This allows to find libnutprivate*.dll or libupsclient*.dll files we built and installed to EXE bindir, which were otherwise ignored by earlier path discovery tools and/or logic.
- scripts/Windows/dllldd.sh: filter_away_NUT_DLLs() when we query both dllldd_with_tools() and dllldd_with_strings() [#3420]
- scripts/Windows/dllldd.sh: refactor with discover_COMPILER_PATHS() to not do this in a loop [#3420]

2.229 2026-05-09 Jim Klimov <jimklimov+nut@gmail.com>

- docs/nut.dict: Update nut.dict [#3430]

2.230 2026-05-08 ostrich <570911+ostrich@users.noreply.github.com>

- scripts/python/app/NUT-Monitor-py3qt5.in, scripts/python/app/NUT-Monitor-py3qt6.in: Fix NUT-Monitor tray tooltip text

2.231 2026-05-08 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: update about NUT-Monitor tooltips [#3430]

2.232 2026-05-07 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsmon.c, NEWS.adoc: do not abort if we did not succeed upscli_init() but did not require SSL in the first place [#3420]
- scripts/Windows/dllldd.sh: refactor to find DLLs with strings too; track filenames we already investigated [#3420] DRY the loop and avoid cycles.

2.233 2026-05-06 Jim Klimov <jimklimov+nut@gmail.com>

- m4/nut_check_libnetsnmp.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libavahi.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- ci_build.sh, scripts/Windows/build-mingw-nut.sh: get_CI_CACHE_NUT_HASHDIR_CFG_OPT(): consider also MAKE and (CONFIG_)SHELL as circumstances that impact the configuration hash [#3108]

2.234 2026-05-05 Jim Klimov <jimklimov+nut@gmail.com>

- m4/nut_check_libfreeipmi.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libgd.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libglib.m4, configure.ac: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libgpod.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libmodbus.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libneon.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libltdl.m4: do not drag around nut_have_libltdl in cached section, use nut_cv_have_libltdl right away [#3108]
- m4/nut_check_libopenssl.m4, m4/nut_check_libnss.m4, configure.ac: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libpowerman.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_libusb.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- m4/nut_check_os.m4: fix comments starting a line with a hash to use "dnl" first

2.235 2026-05-04 Jim Klimov <jimklimov+nut@gmail.com>

- m4/nut_check_libregex.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
 - m4/nut_check_libsystemd.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
 - m4/nut_check_libwrap.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
 - m4/nut_check_pkgconfig.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
 - m4/nut_check_os.m4: set OS_NAME and PKG_TARGET after passing AC_CACHE_CHECK [#3108]
 - m4/nut_check_libltdl.m4: do not reset nut_have_libltdl=yes when checking for header from cached data [#3108]
 - m4/nut_check_libltdl.m4: add a summary print-out for cached re-runs [#3108]
-

- common/common.c: upsnotify(): fix "unused" warnings in different build combos
- ci_build.sh, ci_build.adoc: when we DO_USE_NUTCI_CACHE_DEBUG, stash also the help text from configure script [#3108]
- m4/nut_check_pkgconfig.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libwrap.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libusb.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libsystemd.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libregex.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libpowerman.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libopenssl.m4, m4/nut_check_libpowerman.m4: m4/nut_check_libopenssl.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libnss.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libnetsnmp.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libneon.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libmodbus.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libgpiod.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libglib.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libgd.m4: fix indentations, use AS_IF and AS_CASE more [#3108]
- m4/nut_check_libfreeipmi.m4: fix indentations, use AS_IF [#3108]
- m4/nut_check_libavahi.m4: fix indentations, use AS_IF [#3108]
- m4/nut_check_cppcheck.m4: fix indentations, use AS_IF and AC_CACHE_VAL [#3108]
- m4/nut_check_bool.m4: fix indentations [#3108]
- m4/nut_check_aspell.m4: fix indentations, use AS_IF [#3108]
- m4/nut_check_asciidoc.m4: fix indentations, use AS_IF [#3108]
- m4/nut_check_libltdl.m4: fix indentations, use AS_IF for outer check [#3108]
- m4/nut_check_os.m4: fix typo in heading
- m4/nut_check_libltdl.m4: revise with AC_CACHE_VAL() for each option to be remembered [#3108]
- Makefile.am: when we touch -d "-10 seconds", hide the errors (such request is known to be not too portable)
- ci_build.sh, scripts/Windows/build-mingw-nut.sh: facilitate re-runs with changed **.m4/.am/.ac sources and enabled autoconf cache: handle precious CCACHE_ args/vars** [#3108]
- ci_build.adoc: clarify some points about caching [#3108]

2.236 2026-05-02 Jim Klimov <jimklimov+nut@gmail.com>

- ci_build.sh, ci_build.adoc: introduce DO_USE_AUTOCONF_CACHE_DEBUG=each mode [#3108]

2.237 2026-05-01 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: move entry for #3302 into v2.8.6 release
-

2.238 2026-04-30 Jim Klimov <jimklimov+nut@gmail.com>

- docs/config-prereqs.txt: do not pull valgrind into default Debian/Ubuntu footprint, it is not ubiquitous Revise other suggestions a bit too

2.239 2026-04-29 Jim Klimov <jimklimov+nut@gmail.com>

- tests/cpptest-client.cpp: revise reporting of UPSCLI_SSL_CAPS* [#3331, #1711]
- clients/nutclient.cpp, clients/nutclient.h, clients/upsclient.c, clients/upsclient.h, docs/man/upscli_ssl_caps.txt: upsclient, nut-client, docs: introduce UPSCLI_SSL_CAPS_CERTHOST* [#3331]
- tests/NIT/nit.sh: consider if we can not do CERTHOST validation [#3331]
- clients/upsclient.c: report if we can not do CERTHOST(name) validation [#3331]
- Makefile.am: .libs-dev-PATH: fix typo

2.240 2026-04-28 Jim Klimov <jimklimov+nut@gmail.com>

- clients/Makefile.am, common/Makefile.am, scripts/obs/debian.rules, scripts/obs/nut.spec, tools/nut-scanner/Makefile.am: **/Makefile.am, scripts/obs/**: bump libnutclient ABI level [#1290, #3331]
- clients/nutclient.cpp: drop obsolete FIXME comments [#3331]
- clients/nutclient.cpp: with openssl builds, do not require non-trivial certhost_name IF we do not have methods to check it (openssl too old) [#3331]
- tests/NIT/nit.sh: refactor by use of check_NIT_certs_NSS() [#1711]
- tests/NIT/nit.sh: introduce check_NIT_certs_NSS() [#1711]
- tests/cpptest-client.cpp: enable testing for CERTHOST with OpenSSL builds, update argument order for NSS builds [#3331, #1711]
- clients/Makefile.am, clients/nutclient.cpp, clients/nutclient.h, docs/man/libnutclient_tcp.txt: clients/nutclient.{cpp,h}, docs, Makefile.am: fix SSLConfig_NSS related argument order to place certhost after certident, to match logical proximity and OpenSSL equivalents; reset ABI age (not backwards compatible to recent v2.8.5 release) [#3331]
- tests/NIT/nit.sh: refactor by use of check_NIT_certs_NSS() [#1711, #3390, #3108]
- tests/NIT/nit.sh: introduce check_NIT_certs_NSS() [#1711]
- tests/NIT/nit.sh: revise sanity-check, not all NSS versions had a *.txt file [#3331, #1711]
- tests/NIT/nit.sh: consider WITH_SSL_CLIENT_CERTIDENT=none to (re)set certificate nickname verification [#3331, #1711]
- tests/NIT/nit.sh: reword SSL support messages [#1711]
- tests/NIT/nit.sh: revise sanity-check, not all NSS versions had a *.txt file [#3331, #1711]

2.241 2026-04-27 Jim Klimov <jimklimov@gmail.com>

- clients/nutclient.cpp: fix NULL⇒nullptr [#3331]
 - tests/NIT/nit.sh: isTestablePython(): fix detection of interpreter shebang via /usr/bin/env [#1711]
 - tests/NIT/nit.sh: isTestablePython(): fix detection of interpreter shebang via /usr/bin/env [#1711]
 - tests/NIT/nit.sh: refactor and improve portability with somehash_filter()/somehash_files() method detection [#3390]
 - ci_build.sh: refactor and improve portability with somehash_filter()/somehash_files() method detection [#3108]
-

2.242 2026-04-27 Jim Klimov <jimklimov+nut@gmail.com>

- .circleci/config.yml: allow various cache contents for same branch/job iterations [#3390]
- tests/NIT/nit.sh: introduce check_NIT_certs() to ensure availability of all expected certs for NSS or OpenSSL scenarios [#3390]
- Makefile.am: .libs-dev-PATH: adjust for MacOS [#1711]
- tests/cpptest-client.cpp: report also NUT_PORT [#1711, #3331]
- tests/cpptest-client.cpp: fix reporting of incoming envvars [#1711, #3331]
- clients/upsclient.c, clients/nutclient.cpp: accept systems where openssl does not define X509_V_ERR_HOSTNAME_MISMATCH [#3331]
- clients/upsclient.c, clients/nutclient.cpp: use pragmas to hide unreachable code warnings when built against openssl without X509 checking methods we use [#3331]
- ci_build.sh, tests/NIT/nit.sh: introduce DO_USE_NUTCI_CACHE_DEBUG to stash optional items not needed in production use of the cache, but rather to track its regressions on re-runs etc. [#3108]
- Makefile.am: use a more portable date spec in touch for start of Epoch
- configure.ac: properly use AC_CACHE_VAL for BSDKVMPROCLIBS [#3108]
- m4/nut_check_libltdl.m4: re-detect ltdl.h details if we needed to retry, to not lose track [#3108]
- tests/NIT/nit.sh: fix NUT_CAPATH for PERL SSL tests on Windows (MSYS2) yet some more [#1711]
- tests/NIT/nit.sh: fix NUT_CAPATH for PERL SSL tests on Windows (MSYS2) with forward slashes too [#1711]
- tests/NIT/nit.sh: fix NUT_CAPATH for PERL SSL tests on Windows (MSYS2) with forward slashes too [#1711]
- tests/NIT/nit.sh: fix portability for recursive "cp" [#3390]

2.243 2026-04-26 Jim Klimov <jimklimov+nut@gmail.com>

- m4/nut_check_libnetsnmp.m4: re-detect libnetsnmp details in mingw builds to not lose track [#3108]
- scripts/Windows/build-mingw-nut.sh: CI_CACHE_NUT_HASHDIR_CFG and configure_nut(): stash a copy of first-run config.log and config.h as examples of resulting configuration [#3108]
- scripts/Windows/build-mingw-nut.sh: add © into the heading
- ci_build.sh: get_CI_CACHE_NUT_HASHDIR_CFG_OPT(), configure_nut(): stash a copy of first-run config.log and config.h as examples of resulting configuration [#3108]

2.244 2026-04-25 Jim Klimov <jimklimov+nut@gmail.com>

- docs/download.txt: fix NOTE block style
 - ci_build.sh, Makefile.am: revise passing CI_CACHE_NUT_HASHDIR_CFG_OPT into distcheck runs [#3108]
 - ci_build.sh: consider_cleanup_shortcut(): do not fail if there is no out-of-tree CI_BUILDDIR/Makefile.in - normally it should not be there (but do check for one in SCRIPTDIR)
 - ci_build.sh: refactor get_CI_CACHE_NUT_HASHDIR_CFG_OPT() into a method, handle distcheck builds too [#3108]
 - NEWS.adoc: update about NIT caching of test certificates [#3390]
 - tests/NIT/README.adoc, docs/nut.dict: document caching [#3108, #3390, #1711]
-

- appveyor.yml, .circleci/config.yml: generalize to DO_USE_NUTCI_CACHE, not just DO_USE_AUTOCONF_CACHE [#3108, #3390]
- ci_build.adoc, docs/nut.dict: document caching [#3108]
- ci_build.sh, tests/NIT/nit.sh: generalize DO_USE_NUTCI_CACHE so AUTOCONF and NIT_TESTCERT toggles may be separate; drop old logic for config.cache in build area; rename CLEAN vars to ..._BEFORE [#1711, #3390]
- tests/NIT/nit.sh: handle DO_CLEAN_AUTOCONF_CACHE [#3390]
- tests/NIT/nit.sh: fix preparation of PEM and JKS client key/cert from OpenSSL [#1711, #3390] Move code that landed into a wrong case.
- tests/NIT/nit.sh: refactor preparation of PEM and JKS server key/cert from OpenSSL [#1711, #3390] Handle a situation with new openssl defaults and old(er) java keytool.
- tests/NIT/Makefile.am, tests/NIT/nit.sh: name the default WITH_SSL_TESTS=best-effort behavior for clarity [#1711]
- ci_build.sh, scripts/Windows/build-mingw-nut.sh: ci_build.sh: name CI_CACHE_NUT_HASHDIR with AUTOCONF prefix [#3108]
- tests/NIT/nit.sh: prepare_NIT_certs(): rename TESTCERT_VARS.sh to .env, generate one [#3390]

2.245 2026-04-24 Jim Klimov <jimklimov+nut@gmail.com>

- tests/NIT/nit.sh: fix NUT_CAPATH for PERL SSL tests on Windows (MSYS2) [#1711]
- tests/NIT/nit.sh: prepare_NIT_certs(): for caching, also generate NSS DB from OpenSSL PEM if we can [#3390]
- tests/NIT/nit.sh: prepare_NIT_certs(): drop obsolete comment [#3390]
- tests/NIT/nit.sh: prepare_NIT_certs(): extend validity of generated test certificates to 20 years [#3390]
- tests/NIT/nit.sh: prepare_NIT_certs(): add generation of JKS (and more PEM) from NSS, but only if we plan to cache the result [#3390]
- tests/NIT/nit.sh: prepare_NIT_certs(): wrap long command lines [#3390]
- tests/NIT/nit.sh: refactor mock cert population logic as prepare_NIT_certs(), and allow caller-provided or automatically cached mocks [#3390]
- appveyor.yml: tweak notification triggers again [#1400]
- tests/NIT/nit.sh: fix NUT_CAPATH for PERL SSL tests on Windows (MSYS2) [#1711]
- NEWS.adoc, clients/nutclient.cpp, clients/nutclient.h, clients/upsclient.c, conf/upsmon.conf.sample.in, docs/man/libnutclient_tcp.txt, docs/man/upscli_add_host_cert.txt: clients/{nut,ups}client, docs: make sure CERTHOST values for FORCESSL/CERTVERIFY are used (C++), allow inheriting global setting via -1 [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: in TcpClient, track the applied SSLConfig* vs. build(/run)time ability to use NSS (and/or) OpenSSL via UPSCLI_SSL_CAPS* and consider that in getter preferences [#3331]
- clients/nutclient.cpp, clients/nutclient.h, docs/man/libnutclient_commands.txt, docs/man/libnutclient_tcp.txt: clients/nutclient.{cpp,h}, docs/man/libnutclient_tcp.txt: add optional CERTHOST verification to OpenSSL code path [#3331]

2.246 2026-04-23 Jim Klimov <jimklimov+nut@gmail.com>

- clients/nutclient.cpp: some builds manage to get into WITH_SSL_CXX without WITH_OPENSSL\NWITH_NSS defined [#3331]
- m4/nut_check_libopenssl.m4, clients/upsclient.c, clients/nutclient.cpp, server/netssl.c: openssl_cert_verify_san_name(): support OpenSSL versions before ASN1_STRING_get0_data() and potentially related helpers [#3331]
- ci_build.sh: comment about the life-cycle of config.cache file [#3108]
- ci_build.sh: align defaults for DO_USE_AUTOCONF_CACHE and DO_CLEAN_AUTOCONF_CACHE with existence of CI_CACHE_NUT_BASEDIR [#3108]
- scripts/Windows/build-mingw-nut.sh: add support for CI_TRACE envvar toggle, like in ci_build.sh
- ci_build.sh, scripts/Windows/build-mingw-nut.sh: pass CI_CACHE_NUT_HASHDIR into Windows builds if we want to and can [#3108]
- NEWS.adoc, appveyor.yml: NEWS.adoc: ability to optionally re-use a config.cache file from older runs moved to NUT v2.8.6 [#3108]
- appveyor.yml, .circleci/config.yml: cache the out-of-tree locations for config.cache [#3108]
- ci_build.sh: support reusable config.cache per "uname \+ script config opts" hash [#3108] Now even several build environments sharing a source directory with a generated configure script should be able to safely use their individual caches.
- ci_build.sh: in developer mode (no BUILD_TYPE) utilize make all-quick for less noise
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: refactor SSLConfig_CERTIDENT_NSS to not construct with a separate certstore_pass [#3331] NSS only has one pass phrase (shared for all private key database entries)
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: refactor TcpClient and Socket to pass around SSLConfig(-derived) instances instead of private fields [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: SSLConfig: add getCertHostBy*() methods [#3331]
- clients/nutclient.cpp: clients/nutclient.{cpp,h}: SSLConfig_*::apply(): prepare and hide debug printouts [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: SSLConfig_CERTIDENT fix polymorphic-friendly reference to _certstore [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclientmem.{cpp,h}: use _empty_str in constructors, just to be sure we have a valid reference to an immortal std::string where we expect one [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: SSLConfig*: introduce clone() methods to own data in our sets of CERTHOST/CERTSTORE/CERTIDENT and delete it properly [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: SSLConfig: in the sets of CERTHOST/CERTSTORE/CERTIDENT data, retain polymorphic-friendly pointers instead of mutilated auto-casted objects [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: SSLConfig::addCertHost(): constrain to adding non-trivial entries only [#3331]

2.247 2026-04-22 Jim Klimov <jimklimov+nut@gmail.com>

- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.{cpp,h}: move SSLConfig* method implementations from header to CPP file [#3331]
 - clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.h: refactor SSLConfig_NSS class to use new storage classes [#3331]
 - clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.h: refactor SSLConfig_OpenSSL class to use new storage classes [#3331]
-

- clients/nutclient.h: extend base SSLConfig class to keep track of optional CERTSTORE, CERTIDENT and/or CERTHOST info [#3331]
- clients/nutclient.cpp, clients/nutclient.h: clients/nutclient.h: introduce SSLConfig* classes to keep CERTSTORE locations and credentials, and CERTIDENT and CERTHOST info [#3331]
- clients/nutclient.h: streamline code style for SSLConfig* classes [#3331]
- clients/nutclient.cpp: Socket() constructor should not rely on openssl_cert_verify_data struct to be pre-zeroed [#3331] OBS builds, and later valgrind, confirmed an issue with random decisions and free()'s based on random data in that struct. Apparently C differs from C++ in that aspect?..

2.248 2026-04-21 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/usbhid-ups.c, NEWS.adoc: reconnect_ups(): report success more visibly [#3423] Also fix setting "quiet" driver status when we actually remain in reconnecting state (or rather will soon re-enter it because situation is not fixed)
- clients/nutclient.cpp, clients/upsclient.c, server/netssl.c: openssl_cert_verify_san_name(): revise loop based on return from *printf() [#3331]
- clients/nutclient.cpp, server/netssl.c: fix CI warnings [#3331]
- UPGRADING.adoc, clients/nutclient.cpp, clients/nutclient.h, docs/man/Makefile.am, docs/man/libnutclient_tcp.txt, tests/NIT/nit.sh, tests/cpptest-client.cpp: clients/nutclient.{cpp,h}, tests/cpptest-client.cpp, nit.sh, docs: a CERTHOST address was missing (and subject/nickname is not the host name/url that NSS checks for) [#3331, #1711]
- clients/nutclient.cpp, clients/nutclient.h, docs/man/libnutclient_tcp.txt: clients/nutclient.{cpp,h}: rename "forcssl" to match upsclient.c, and "tryssl" to not stand out [#3331]
- tests/cpptest-client.cpp: honour NUT_DEBUG_LEVEL by enabling (Socket Connect) debug toggle [#1711]
- tests/NIT/nit.sh: support passing individual NUT_DEBUG_LEVEL_{PERL,PYTHON,CPPNIT,NUTSCAN} [#1711]
- server/netssl.c, clients/upsclient.c, clients/nutclient.cpp: align messages from NSS AuthCertificate() callbacks [#3331]
- clients/nutclient.cpp: Socket::startTLS(): if NSS handshake fails, add ssl_url to the exception message [#3331]
- clients/nutclient.cpp: setSSLConfig_*(): debug-log the code path actually taken [#1711]
- clients/nutclient.cpp: Socket: document the SSL fields better [#3331]
- clients/nutclient.cpp: TcpClient::connect(): now we can config run-time for both NSS and OpenSSL, most options are shared and built-in abilities drive the final choice [#3331]
- tests/cpptest-client.cpp: setupClientSSL(): rearrange reporting of enabled SSL options [#1711]

2.249 2026-04-20 Jim Klimov <jimklimov+nut@gmail.com>

- tests/NIT/nit.sh: setenv_ssl_cppnit(): provide NUT_CERTFILE and NUT_KEYFILE values for OpenSSL tests [#1711, #3331]
 - clients/nutclient.cpp: transplant openssl_cert_verify_callback() et al from upsclient.{c,h} [#3331]
 - clients/nutclient.cpp: Socket::startTLS(): some reported CN values (at least from openssl parser) may include escaped slashes as a separator [#3331]
 - clients/nutclient.cpp: fix syntax for NSS builds [#3331, #1711]
 - configure.ac: fix the checks for enabling SSL support in C++ builds [#1711, #3331]
 - server/netssl.c: ssl_init(): some reported CN values (at least from openssl parser) may include escaped slashes as a separator [#3331]
 - server/netssl.c, server/nut_ctype.h: transplant openssl_cert_verify_callback() et al from upsclient.{c,h} [#3331]
-

- clients/upsclient.c, clients/upsclient.h: clients/upsclient.{c,h}: revise code/comment markup for recent changes [#3331]
- tests/NIT/nit.sh: ensure SAN aliases (DNS and IP) that should NOT match the current host [#1711]
- clients/upsclient.c: upscli_init2(): some reported CN values (at least from openssl parser) may include escaped slashes as a separator [#3331]
- clients/upsclient.c: openssl_cert_verify_san_name(): use X509_check_host()/X509_check_ip_asc() where available [#3331]
- clients/upsclient.c: openssl_cert_verify_callback(): consistently report depth along with CN (counterpart, CA...) [#3331]
- clients/upsclient.c: openssl_cert_verify_callback(): do not call and log openssl_cert_verify_san_name() when already OK per caller [#3331]
- clients/upsched.c: hush down byte-by-byte read tracing (away from common debug level 6)
- clients/upsclient.c, clients/upsclient.h: clients/upsclient.c: convert openssl_cert_print_san_name() into openssl_cert_verify_san_name() [#3331] Also persist openssl_cert_verify_data_t in UPSCONN_t so it is always available to callbacks when needed for this ups instance.
- clients/upsclient.c: openssl_cert_print_san_name(): handle IPv4/IPv6 addresses in SAN [#3331]
- clients/upsclient.c: rename example "mydata" into "openssl_cert_verify_data" (types, vars) and example methods to print/verify certs [#3331]
- clients/upsclient.c, configure.ac: import openssl doc examples for certificate reporting/inspection into NUT styling and logging [#3331]
- clients/upsclient.c: fix markup about switch() so it is consistent in the file [#3331]
- clients/nutclient.cpp, clients/upsclient.c, server/netssl.c: server/netssl.c, clients/upsclient.c, clients/nutclient.cpp: when we verify subject_CN[], consider comma as a separator too [#3331]
- clients/nutclient.cpp: fix a missed static_cast [#3331]

2.250 2026-04-19 Jim Klimov <jimklimov+nut@gmail.com>

- clients/nutclient.cpp: be sure to include openssl/x509v3.h for methods we use [#3331]

2.251 2026-04-18 Tom Eldon <tom@tomeldon.com>

- drivers/apcmicrolink.c, drivers/apcmicrolink.h: drivers/apcmicrolink.c: minor handshake fix, allow flexible frame size Change to allow flexible frame size (upto 256 bytes) by waiting for page0, as some devices with payload size != 35 were incompatible. Allow variable length serial number string during handshake (as some devices use 14 characters).

2.252 2026-04-18 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/apc_modbus.c: addvar() the modbus_retries option [#3414]

2.253 2026-04-16 Jim Klimov <jimklimov+nut@gmail.com>

- tests/NIT/nit.sh: setenv_ssl_common(): revise NUT_CAPATH conditions [#1711]
 - tests/NIT/nit.sh: fix check of python FORCESSL is testable [#1711, #1349]
-

2.254 2026-04-15 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/module/PyNUT.py.in: fix sleep() [#1349, #1711]
- clients/nutclient.cpp, clients/nutclient.h, clients/upsclient.c, clients/upsclient.h, docs/man/libnutclient_tcp.txt, docs/man/upscli_ssl_caps.txt, server/netssl.c, tests/NIT/nit.sh: Further differentiate reported CERTIDENT_NAME/_PASS capability [#1711]
- tests/NIT/nit.sh: refactor with setenv_ssl_common(), and in setenv_ssl_python() verify that we can run SSL tests (else do not force them) [#1349, #1711]
- scripts/python/module/PyNUT.py.in: detect inability to use SSL module (older interpreter etc.) [#1349, #1711]
- tests/NIT/nit.sh: align generation of CERTIDENT into upsd.conf/upsmon.conf with ability announced by the binaries [#1711]
- scripts/perl/test_nutclient.pl: ANSI colors are a nice to have, do not crash if absent [#1711]
- scripts/perl/UPS/Nut.pm: StartTLS(): use a better portable "or" for "undef" printing [#1711]
- scripts/perl/test_nutclient.pl: warnings are a nice to have, do not crash if absent [#1711]
- NEWS.adoc, clients/nutclient.cpp, clients/nutclient.h, clients/upsclient.c, clients/upsclient.h, docs/man/libnutclient_tcp.txt, docs/man/upscli_ssl_caps.txt: clients/upsclient.c, clients/nutclient.cpp, headers, docs: propagate UPSCLI_SSL_CAPS_CERTIDENT as a flag [#3331]
- server/netssl.c, clients/upsclient.c: *ssl_caps_descr(): mention OpenSSL CERTIDENT (in)ability [#3331]
- scripts/perl/UPS/Nut.pm: warnings are a nice to have, do not crash if absent [#1711]
- clients/upsclient.c, clients/nutclient.cpp, server/netssl.c, m4/nut_check_libopenssl.m4: for OpenSSL password callback, check actual presence of methods, not version numbers [#3331]
- clients/upsclient.c: nss_password_callback(): reconcile markup with server/netssl.c [#3331]
- server/netssl.c: for now openssl_password_callback() should only be declared/implemented when OPENSSL_VERSION_NUMBER >= 1.1.0 [#3331]
- tests/NIT/nit.sh: account SKIPPED test cases (when isTestable* says we can not) [#1711]
- tests/NIT/nit.sh: be portable, use TABCHAR instead of "\t" [#1711]
- docs/nut.dict: update for older systems
- configure.ac: work around ancient autotools without PACKAGE_URL
- Makefile.am: hijack PARMAKES_OPT to also set V=0 by default

2.255 2026-04-14 Lukas Schmid <lukas.schmid@netcube.li>

- drivers/apcmicrolink-maps.c, drivers/apcmicrolink-maps.h, drivers/apcmicrolink.c: drivers/apcmicrolink.c: try to use naming from apc_modbus for some commands and values
- drivers/apcmicrolink.c: fix microlink_page0_state_t init and reset

2.256 2026-04-14 Jim Klimov <jimklimov+nut@gmail.com>

- clients/upsclient.c: for now openssl_password_callback() should only be declared/implemented when OPENSSL_VERSION_NUMBER >= 1.1.0 [#3331]
 - conf/upsmon.conf.sample.in, docs/man/upsmon.conf.txt: clarify that CERTIDENT support requires OpenSSL 1.1.0\+ to fully work at the moment [#3331]
 - drivers/nutdrv_qx.c, NEWS.adoc: qx_is_usb_device_supported(): only claim SUPPORTED if subdriver_command was assigned [#3410]
-

2.257 2026-04-14 Axel Gembe <axel@gembe.net>

- NEWS.adoc, drivers/apc_modbus.c: apc_modbus: increase default TCP response timeout to 2000 ms The default 500ms timeout value is marginal for Modbus TCP, even on a fast local network. In tests with an APC UPS, we see timeouts because of this: `` Apr 14 19:10:44 ares apc_modbus[2002143]: _apc_modbus_read_registers: Read of 0:27 failed: Connection timed out (192.168.0.102:502) Apr 14 19:10:44 ares apc_modbus[2002143]: _apc_modbus_read_registers: Read of 0:27 failed: Invalid data (192.168.0.102:502) Apr 14 19:10:44 ares apc_modbus[2002143]: _apc_modbus_close: Closing connection Apr 14 19:10:47 ares apc_modbus[2002143]: Opened modbus successfully `` Packet capture: `` No. Time Source Destination Protocol Length Info 3636 2026-04-14 19:10:43.842072 192.168.0.40 192.168.0.102 Modbus/TCP 66 Query: Trans: 12622; Unit: 1, Func: 3: Read Holding Registers 3637 2026-04-14 19:10:44.018369 192.168.0.102 192.168.0.40 TCP 60 502 → 40392 [ACK] Seq=70441 Ack=11905 Win=3660 Len=0 3638 2026-04-14 19:10:44.377723 192.168.0.40 192.168.0.102 Modbus/TCP 66 Query: Trans: 12623; Unit: 1, Func: 3: Read Holding Registers 3639 2026-04-14 19:10:44.578103 192.168.0.102 192.168.0.40 TCP 60 502 → 40392 [ACK] Seq=70441 Ack=11917 Win=3648 Len=0 3640 2026-04-14 19:10:44.579449 192.168.0.102 192.168.0.40 Modbus/TCP 117 Response: Trans: 12622; Unit: 1, Func: 3: Read Holding Registers 3641 2026-04-14 19:10:44.579482 192.168.0.40 192.168.0.102 TCP 54 40392 → 502 [ACK] Seq=11917 Ack=70504 Win=63 Len=0 3642 2026-04-14 19:10:44.579604 192.168.0.40 192.168.0.102 TCP 54 40392 → 502 [FIN, ACK] Seq=11917 Ack=70504 Win=63 Len=0 `` 3636 is the first read, 3638 is the retry after 500ms without response, both are ACKed. 3640 is the actual response to transaction 12622, which arrives 700ms after the initial packet. At this point we expect a response to transaction 12623, which is why we fail with "invalid data". To fix this we increase the default TCP response timeout to 2000 ms, which seems to be more reasonable for a device that already can take 700 ms to respond under ideal conditions. I chose a bit higher value for users who might connect the device over a higher latency network.
- NEWS.adoc, docs/nut.dict, drivers/apc_modbus.c, drivers/apc_modbus.h: apc_modbus: decode RunTimeCalibrationStatus_BF RunTimeCalibrationStatus_BF is explained in MPAO-98KJ7F_R1_EN Appendix B. This adds defines for the bitfield and decodes it into experimental.ups.calibration.result using a string join converter.
- NEWS.adoc, docs/nut.dict, drivers/apc_modbus.c, drivers/apc_modbus.h: apc_modbus: decode PowerSystemError_BF as alarms PowerSystemError_BF is explained in MPAO-98KJ7F_R1_EN Appendix B. This adds defines for the bitfield and decodes it as alarms in the driver.
- NEWS.adoc, docs/nut.dict, drivers/apc_modbus.c, drivers/apc_modbus.h: apc_modbus: decode BatterySystemError_BF as alarms BatterySystemError_BF is explained in MPAO-98KJ7F_R1_EN Appendix B. We were already using the bitfield for the RB status, this adds defines for the bitfield and decodes it as alarms in the driver.
- NEWS.adoc, docs/man/apc_modbus.txt, drivers/apc_modbus.c: apc_modbus: simplify error handling with read retries Replace the complex _apc_modbus_handle_error function with a simpler retry mechanism built into _apc_modbus_read_registers The new approach: - Retries register reads on ETIMEDOUT errors up to modbus_retries times (configurable, default 3) - On non-timeout errors or after retry exhaustion, closes the connection for reconnection on the next update cycle - Removes the platform-specific (WIN32/POSIX) timeout detection and the flush-based recovery that didn't work anyway (flush is already done in _apc_modbus_reopen upon reconnection) Also adds a modbus_retries driver option to configure the number of read retry attempts, and improves logging for connection open/close events. This change was inspired by a patch by @marcan to do the same and from testing the behaviour of apcupsd, noticing that on my USB unit it times out on read and only succeeds on the first retry.
- NEWS.adoc, drivers/apc_modbus.c: apc_modbus: always zero terminate string join _apc_modbus_string_join did not zero terminate the result in every case, like when all values are NULL.
- drivers/apc_modbus.c: apc_modbus: bump driver version to 0.20

2.258 2026-04-13 Lukas Schmid <lukas.schmid@netcube.li>

- docs/nut.dict: Add more words to dict
- drivers/apcmicrolink.c: fix most static compile issues
- docs/nut.dict: add apcmicrolink and RMI

2.259 2026-04-13 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: introduce apcmicrolink driver [#3406]
 - drivers/Makefile.am: EXTRA_DIST apcmicrolink-maps.h apcmicrolink.h [#3406]
 - docs/man/Makefile.am: provide apsmicrolink.html also [#3406]
 - tests/NIT/nit.sh: generate upsmon.pem [#3331, #1711]
 - NEWS.adoc, clients/nutclient.cpp, clients/upsclient.c, conf/upsd.conf.sample, conf/upsmon.conf.sample.in, docs/man/upsd.conf.txt, docs/man/upsmon.conf.txt, server/netssl.c: upsd, libupsclient, libnutclient, upsmon: in OpenSSL builds, support CERTPATH pointing to a big PEM file as well as a directory [#3331]
 - clients/nutclient.cpp, clients/nutclient.h, tests/cpptest-client.cpp: clients/nutclient.{h,cpp}, tests/cpptest-client.cpp: extend setSSLConfig_OpenSSL() etc. to support CERTIDENT name (validation of client' own cert) [#3331]
 - clients/upsclient.c: fix CERTIDENT (OpenSSL) to check for CN=... part too [#3331]
 - server/netssl.c: fix CERTIDENT (OpenSSL) to check for CN=... part too [#3331]
 - tests/NIT/nit.sh: generate upsmon.pem [#3331, #1711]
 - tests/NIT/nit.sh: setenv_ssl_cpptest(): fix check for TESTCERT_PATH_ROOTCA [#1711]
 - tests/NIT/nit.sh: unlock OpenSSL features that are now on par with NSS [#3331, #1711]
 - scripts/python/module/PyNUT.py.in, scripts/perl/UPS/Nut.pm: handle CERTIDENT to validate loaded client certificate name [#3331] Make sure the file we got is the file we wanted. Belts and suspenders rule!
 - scripts/python/module/PyNUT.py.in, scripts/perl/UPS/Nut.pm: handle CERTHOST (dict or list of dict/tuple) to validate connections better [#3331]
 - scripts/perl/UPS/Nut.pm: StartTLS(): alias variables named like in other NUT code into SSL_* if not provided directly [#1711]
 - clients/nutclient.cpp, clients/upsclient.c, conf/upsd.conf.sample, docs/man/upsd.conf.txt, server/netssl.c: clients/nutclient.cpp, clients/upsclient.c, server/netssl.c: reconcile variants of openssl_password_callback() [#3331] Notably, in C code (upsclient, netssl) constrain with OpenSSL >= 1.1 support, and do use SSL_CTX_set_default_passwd_cb_userdata() to pass the configured (or generally otherwise provided) certpasswd string.
 - NEWS.adoc, clients/upsmon.c, conf/upsmon.conf.sample.in, docs/man/upsmon.conf.txt: clients/upsmon.c, NEWS.adoc, conf/, **docs/man/**: introduce support of CERTFILE to upsmon [#3331]
 - clients/upsclient.c, clients/upsclient.h, docs/man/Makefile.am, docs/man/upscli_init.txt: clients/upsclient.{c,h}, docs/man/upscli_init.txt: implement client CERTIDENT correctly (via CERTFILE); introduce upscli_init2() [#3331]
 - NEWS.adoc, conf/upsd.conf.sample, docs/man/upsd.conf.txt, server/conf.c, server/netssl.c, server/netssl.h: server/*, NEWS.adoc, conf/upsd.conf.sample, docs/man/upsd.conf.txt: implement server-side CERTPATH for OpenSSL [#3331] Optionally use for validation of clients with OpenSSL collection of CA PEM files.
 - conf/upsd.conf.sample, docs/man/upsd.conf.txt, server/conf.c, server/netssl.c: server/*, conf/upsd.conf.sample, docs/man/upsd.conf.txt: implement server-side CERTIDENT for OpenSSL [#3331]
 - NEWS.adoc, conf/upsd.conf.sample, conf/upsmon.conf.sample.in, docs/man/upsd.conf.txt, docs/man/upsmon.conf.txt: NEWS.adoc, docs/man/, **conf/**: update docs about newly supported features with OpenSSL [#3331]
 - server/conf.c: reshuffle options processing and update comments now that CERTREQUEST is for both backends and more may follow [#3331]
 - clients/upsclient.c: upscli_sslinit(): consider CERTHOST→certverify setting to NOT abort the connection if verification was not required (0) [#3331]
 - clients/upsclient.c: upscli_sslinit(): consider OPENSSL_VERSION_NUMBER constraint in logging, possibly rejecting the StartTLS attempt with OpenSSL too old (can not fulfil required security, avoid sad mishaps) [#3331]
-

- `conf/upsd.conf.sample`, `docs/man/upsd.conf.txt`: heads-up about MAXCONN and ulimit [#3365]
- `NEWS.adoc`: remove duplicated paragraph (silent merge conflict)
- `clients/upsmon.c`, `clients/upsrw.c`, `common/common.c`, `drivers/adelsystem_cbi.c`, `drivers/generic_modbus.c`, `drivers/main.c`, `drivers/powerman-pdu.c`, `drivers/snmp-ups.c`, `drivers/tripplite_usb.c`, `tools/nut-scanner/nutscan-serial.c`: **.c: fix use of non-const char** with `strstr()` and `strchr()` return values [#823]
- `docs/man/Makefile.am`: fix possible non-delivery of NUTSCAN_UPSLOG_SET_DEBUG_LEVEL_DEPS pages [#3378, #3379]
- `docs/man/upscli_init.txt`: update wording and markup for pre-existing features [#3331]
- `NEWS.adoc`: announce the adjusted markup format

2.260 2026-04-12 Jim Klimov <jimklimov@gmail.com>

- `scripts/perl/UPS/Nut.pm`: warn if CERTVERIFY and custom CA options are enabled on Darwin platform [#3404]
- `scripts/perl/UPS/Nut.pm`: align closer with `can_ssl` [#1711]
- `scripts/perl/UPS/Nut.pm`, `scripts/perl/test_nutclient.pl`, `tests/NIT/nit.sh`: `scripts/perl/*`, `tests/NIT/nit.sh`: streamline passing `NUT_DEBUG` (or defaulting to no debug) [#1711]
- `tests/NIT/nit.sh`: neuter CERTVERIFY on darwin for now [#3404]
- `tests/NIT/nit.sh`: centralize definition of `PERL_OPTS` [#1711]
- `scripts/perl/UPS/Nut.pm`: allow SSL cert validation for IP addresses as host names [#1711]
- `scripts/perl/UPS/Nut.pm`: pass only defined args to `IO::Socket::SSL→start_SSL()` [#1711]
- `scripts/perl/UPS/Nut.pm`: support debugging messages of `IO::Socket::SSL` [#1711]
- `tests/NIT/nit.sh`: report the non-default `NIT_CASE` being handled [#1711]

2.261 2026-04-12 Jim Klimov <jimklimov+nut@gmail.com>

- `clients/upsclient.c`: `upscli_sslinit()`: `OPENSSL_VERSION_NUMBER` constraint applies to whole block for `X509_VERIFY_PARAM` [#3331]
 - `server/netssl.c`: extend OpenSSL builds feature set to support CERTREQUEST validation of clients [#3331]
 - `clients/upsclient.c`: extend OpenSSL builds feature set to support pinning server certificates [#3331]
 - `clients/upsclient.c`: extend OpenSSL builds feature set to support a client certificate\+key, optionally with password [#3331]
 - `scripts/perl/UPS/Nut.pm`, `scripts/perl/test_nutclient.pl`: `scripts/perl/UPS/Nut.pm`: now we can pass SSL debug level as a number [#1711]
 - `scripts/perl/UPS/Nut.pm`: fix to loading `IO::Socket::SSL` via `eval` with quotes not braces [#1711] It is then evaluated at run-time as code reaches that line (or not - conditionals), not at interpretation time where it fails too early. Also use `"1;"` in the end as a tidy practice.
 - `clients/nutclient.cpp`: work around lack of threads in some mingw versions [#3402]
-

2.262 2026-04-11 Lukas Schmid <lukas.schmid@netcube.li>

- data/driver.list.in, docs/man/Makefile.am, docs/man/apcmicrolink.txt, drivers/Makefile.am, drivers/apcmicrolink-maps.c, drivers/apcmicrolink-maps.h, drivers/apcmicrolink.c, drivers/apcmicrolink.h: drivers: add experimental APC Microlink serial driver Implement a new apcmicrolink driver for APC Smart-UPS units that expose the Microlink serial protocol. The driver establishes a Microlink session over the serial link, reads the device descriptor blob, and uses that runtime descriptor map to publish standard NUT variables, writable settings, status bits, alarms, and supported instant commands. Add labeled Microlink command masks for battery tests, calibration, UPS control, and outlet-group power control; wire APC-style load and shutdown commands through the static command map; publish input.transfer.reason from UPS status change cause; and stop folding simple-signaling status bits into ups.status. Also hook upsdrv_shutdown() up to the Microlink shutdown request, document the driver, and clean up initial upstream review issues such as missing file headers, whitespace, and format-string warnings.

2.263 2026-04-11 Jim Klimov <jimklimov+nut@gmail.com>

- m4/ax_c_pragmas.m4, clients/nutclient.cpp: introduce HAVE_PRAGMA_GCC_DIAGNOSTIC_IGNORED_DEPRECATED_DECLARATIONS to implement methods for deprecated "master" operation [#840, #3402]

2.264 2026-04-10 Jim Klimov <jimklimov+nut@gmail.com>

- NEWS.adoc: announce updates to client binding libraries [#3402]
- scripts/perl/test_nutclient.pl: avoid "Can't call method "Error" on an undefined value" messages in failed tests; clearly say that \$nut is no more [#1711]
- scripts/perl/UPS/Nut.pm: when IO::Socket::SSL is not available, report the error with "FEATURE-NOT-SUPPORTED" in text [#1711]
- tests/NIT/nit.sh: setenv_ssl_perl(): check if IO::Socket::SSL is available, neuter NUT_FORCESSL based on that [#1711]
- scripts/perl/test_nutclient.pl: support USESSL=undef if no value provided in env [#1711]
- scripts/perl/UPS/Nut.pm: support USESSL=undef, check that we can_ssl and follow up with that [#1711]
- scripts/perl/UPS/Nut.pm: _initialize(): include previously reported error in SSL setup fail messages [#1711]
- scripts/perl/UPS/Nut.pm: apply indentation consistently (3 styles were intermixed)
- scripts/perl/UPS/Nut.pm, scripts/perl/test_nutclient.pl: support NUT_SSL envvar toggle to disable attempts at SSL altogether [#1711]
- scripts/perl/UPS/Nut.pm, scripts/perl/test_nutclient.pl: SSL_ca_path should be defined in args by caller, not by a hack in production code [#1711]
- scripts/perl/UPS/Nut.pm: STARTTLS: debug-print %arg list that is passed to the SSL library [#1711]
- scripts/perl/UPS/Nut.pm: use CAPATH to provide SSL_ca_path, not SSL_ca_file [#1348, #1711]
- scripts/python/module/PyNUT.py.in: retry SSL handshake if it timed out, a few times [#3401]
- server/netssl.c: net_starttls(): report which SSL backend is in place (if any) [#3331]
- scripts/python/module/PyNUT.py.in: isValidProtocolVersion(): stringify and strip the "result" before matching in regex [#1349]
- clients/Makefile.am: bump libupsclient ABI version
- tests/NIT/nit.sh: add a note about jNut using this script [#1711, #1350]
- clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.cpp, clients/nutclientmem.h, tests/cpptest-client.cpp: clients/nutclient{,mem}.{h,cpp}, tests/cpptest-client.cpp: remember if we enabled TRACKING so it is really done once, confirm with isTrackingModeEnabled() [#656]

- scripts/perl/test_nutclient.pl: fix DeviceLogin⇒ListClient for "admin" without a role [#1711]
- scripts/perl/UPS/Nut.pm: _send(): reset the error buffer before networking away [#1711, #1348]
- scripts/perl/UPS/Nut.pm: we can only authenticate once per connection, skip other attempts [#1711, #1348]
- docs/download.txt: publish jNut-1.1 [#1350]
- NEWS.adoc: announce updates to client binding libraries [#3402]
- appveyor.yml: post only one (custom) notification per build [#1400]
- docs/man/Makefile.am: hush "cat" errors in LIBNUTCLIENT_TCP_DEPS_SUB recipe

2.265 2026-04-09 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/perl/test_nutclient.pl: fix reporting of @failed methods [#1711]
- scripts/perl/UPS/Nut.pm: GetTrackingResult(): fix parsing results of GET TRACKING [#1348]
- tests/NIT/nit.sh: when we ask for specific NIT_CASE values (cppnit, python, perl, nutscanner) the lack of prerequisites should be a FAILURE [#1711]
- tests/NIT/nit.sh: introduce tests for PERL [#1348, #1711]
- message, so we can parse it [#1348, #1711]
- scripts/perl/test_nutclient.pl: fix access to possibly undefined envvars [#1348, #1711]
- scripts/perl/UPS/Nut.pm: fix error messages when Authenticate() and Login() fail [#1348, #1711]
- scripts/perl/UPS/Nut.pm, scripts/perl/test_nutclient.pl: scripts/perl/test_nutclient.pl: fix test case setup, update comments [#1711, #1348]
- scripts/perl/UPS/Nut.pm: fix string work with TRACKING logic
- scripts/perl/UPS/Nut.pm: change commands that address an UPS to optionally accept the UPS name as argument, only using
- scripts/python/module/test_nutclient.py.in: fix typo in debug traces
- assignments into debug() calls [#1348, #1711]
- COPYING, scripts/HP-UX/nut.psf.in, scripts/Makefile.am, scripts/obs/debian.libups-nut-perl.install, scripts/perl/{ ⇒ UPS}/Nut.pm, scripts/perl/test_nutclient.pl: scripts/perl/test_nutclient.pl: introduce tests, move Nut.pm to UPS subdir [#1348, #1711]
- scripts/perl/Nut.pm: call StartTLS() from _initialize(), and use isValidProtocolVersion() method to actively ping a connection after STARTTLS claimed success (but handshake might be broken in fact) [#3387, #1348]
- clients/nutclient.cpp, clients/upsclient.c, scripts/python/module/PyNUT.py.in: Client libraries (C, C+++, Python): use isValidProtocolVersion() method to actively ping a connection after STARTTLS claimed success (but handshake might be broken in fact) [#3387]
- clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.cpp, clients/nutclientmem.h, clients/upsclient.c, clients/upsclient.h, scripts/perl/Nut.pm, scripts/python/module/PyNUT.py.in: Client libraries (C, C+++, Python, PERL): introduce an isValidProtocolVersion() method to actively ping a connection [#3387]
- clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.cpp, clients/nutclientmem.h: clients/nutclient{,mem}.{h,cpp}: becomeSecondary() was overkill, there is no such verb in NUT protocol (LOGIN already reaches that role level)
- scripts/python/module/PyNUT.py.in: refactor FSD() with an explicit becomePrimary() method
- scripts/perl/Nut.pm: becomeSecondary() was overkill, there is no such verb in NUT protocol (LOGIN already reaches that role level)
- scripts/perl/Nut.pm: add warnings toggle

- scripts/perl/Nut.pm: fix typos in comments
- scripts/perl/Nut.pm: rename primary() $\Rightarrow$ becomePrimary, add becomeSecondary() for completeness [#1348]
- clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.cpp, clients/nutclientmem.h: clients/nutclient{,mem}.{h,cpp}: rename primary() $\Rightarrow$ becomePrimary, add becomeSecondary() for completeness [#656]
- tests/cpptest-client.cpp: add test with SET VAR TRACKING [#1711, #656]
- scripts/python/module/test_nutclient.py.in: add test with SET VAR TRACKING [#1711, #1349]
- scripts/perl/Nut.pm: fix typos in comments
- docs/download.txt: refer to new release of jNut-1.0 (and the following snapshot evolution)
- tests/NIT/nit.sh: do not unset DEBUG_SLEEP when we generate configs [#1711] We might actually want more of those, with test case passwords revealed, etc.

2.266 2026-04-08 Jim Klimov <jimklimov+nut@gmail.com>

- scripts/python/module/PyNUT.py.in: TrackingID class: include timestamp tracking and age reporting [#1349]
 - scripts/perl/Nut.pm: TrackingID class: include timestamp tracking and age reporting [#1348]
 - clients/nutclient.cpp, clients/nutclient.h, clients/nutclientmem.cpp, clients/nutclientmem.h: clients/nutclient{,mem}.{h,cpp}, tests/*.cpp: implement waiting for TRACKING result if requested [#656]
 - clients/nutclient.h, clients/Makefile.am: Implement TrackingID as a separate class, including timestamp tracking and age reporting [#656]
 - scripts/python/module/PyNUT.py.in: Implement TrackingID as a separate class [#1349]
 - scripts/python/module/PyNUT.py.in: fix syntax [#1349]
 - scripts/perl/Nut.pm: Implement TrackingID as a separate class [#1348]
 - scripts/perl/Nut.pm: fix syntax [#1348]
 - scripts/python/module/PyNUT.py.in: revise AI creations, complete the TRACKING\+waiting dialog support [#1349]
 - scripts/perl/Nut.pm: revise AI creations, complete the TRACKING\+waiting dialog support [#1348]
 - scripts/perl/Nut.pm, scripts/python/module/PyNUT.py.in: AI PoC: implement the remainder of NUT Networked protocol (as of NUT v2.8.5) for Python and Perl bindings
 - scripts/perl/Nut.pm: clean up trailing white-space
 - tests/NIT/nit.sh: simplify with NIT_CASE=generatecfg_sandbox support [#1711]
 - tests/NIT/nit.sh: allow injecting WITH_SSL_CLIENT, WITH_SSL_SERVER, WITH_SSL_SERVER_CLIVAL (e.g. to test against older packaged builds) [#1711]
 - .github/workflows/05-codeql.yml: drop duplicate init stage
 - configure.ac: hush clang-21 "-Wc11-extensions" warnings about system headers
 - docs/download.txt: fix NOTE markup for N4W 7Zip size
-

2.267 2026-04-07 Jim Klimov <jimklimov+nut@gmail.com>

- docs/maintainer-guide.txt: revise instructions for post-release steps (website, etc.)
- docs/download.txt: newly refer to nut-X.Y.Z-docs.tar.gz archives
- docs/docinfo.xml.in: fix release date of v2.8.5 and move its tag to be above v2.8.4
- NEWS.adoc, UPGRADING.adoc, docs/docinfo.xml.in: Revert "NEWS.adoc, UPGRADING.adoc, docs/docinfo.xml.in: finalize text before NUT v2.8.5 release" This reverts commit 784219a3dc935e53a1273063c591bc6dd949590b: finally released NUT v2.8.5, preparing for v2.8.6 cycle now.
- tools/gitlog2version.sh: for development, set fallback NUT_VERSION_DEFAULT=2.8.5.1

2.268 2026-04-06 Jim Klimov <jimklimov+nut@gmail.com>

- common/wincompat.c: syslog(), send_to_named_pipe(): debug-log when we failed to open existing NAMED_PIPE and so move on [#3302]
- common/wincompat.c: log errors after methods faults via upslog_with_errno() not plain upslogx() [#3302]
- common/wincompat.c: log progress through pipe*() methods [#3302, #3368]
- common/wincompat.c: fix send_to_named_pipe() to measure strlen(data) once [#3302]
- common/wincompat.c: fix getpass() to measure strlen(wincompat_password) once [#3302]
- common/wincompat.c: fix filter_path() to measure strlen(source) once [#3302]
- common/wincompat.c: tcflush(): fix bitwise OR to boolean OR

2.269 2026-03-31 Jim Klimov <jimklimov+nut@gmail.com>

- server/upsd.c: sendback(): adjust debug traces for write() vs ssl_write() [#3379, #3331]
- server/netssl.c: net_starttls(): send_err_extra(NUT_ERR_ACCESS_DENIED) for faults in NSS setup [#3379, #3331]
- server/upsd.c, server/upsd.h: server/upsd.{c,h}: introduce send_err_extra() [#3331]
- drivers/dstate.c, NEWS.adoc: revise closing of connections and returning error codes; document the methods [#3302]

2.270 2026-03-30 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/dstate.c: debug-log the progress around starting and maybe failing CreateNamedPipe() [#3302]
- drivers/dstate.c: sock_disconnect(): resync code with wincompat::pipe_disconnect() implementation [#3302]
- drivers/dstate.c: sock_disconnect(): try to flush the buffer [#3368]

2.271 2026-03-25 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/dstate.c: err on the safe side and set conn=NULL after sock_disconnect() [#3368]

2.272 2026-03-24 Jim Klimov <jimklimov+nut@gmail.com>

- drivers/upsdrvctl.c: refactor with forkexec_parent_analyze() and a re-check before retry - also for WIN32 [#3302] Also revised WaitForSingleObject() result checking - there has to be a chance to succeed ;)
 - drivers/upsdrvctl.c: refactor with forkexec_parent_analyze() and a re-check before retry - POSIX part [#3302]
-

2.273 2025-10-02 Jim Klimov <jimklimov+nut@gmail.com>

- configure.ac: WIN32 GetAdaptersInfo(): be careful about \$NETLIBS_GETADDRS usage
 - ci_build.sh: retain config.cache if DO_CLEAN_AUTOCONF_CACHE=no also in dev-test builds [#3108]
 - ci_build.sh: configure_nut(): log if config.cache is available and if we want/do use it [#3108]
 - ci_build.sh: do not default to DO_USE_AUTOCONF_CACHE=yes/DO_CLEAN_AUTOCONF_CACHE=no [#3108] Callers like specific CI recipes should be free to enable certain behaviors though.
 - .circleci/config.yml: default to DO_CLEAN_AUTOCONF_CACHE:no DO_USE_AUTOCONF_CACHE:yes and carry over the config.cache between builds [#3108]
 - appveyor.yml: default to DO_CLEAN_AUTOCONF_CACHE:no DO_USE_AUTOCONF_CACHE:yes and carry over the config.cache between builds [#3108]
 - ci_build.sh, scripts/Windows/build-mingw-nut.sh, NEWS.adoc: introduce options to USE_AUTOCONF_CACHE and/or CLEAN_AUTOCONF_CACHE [#3108]
-