

Workgroup: IETF
Internet-Draft:
draft-rprice-ups-management-protocol-02
Published: 6 March 2021
Intended Status: Informational
Expires: 7 September 2021
Authors: R. Price, Ed.

Network UPS Tools Project
Uninterruptible Power Supply (UPS) Management Protocol -- Commands and Responses

Abstract

This text describes the command/response protocol currently used in the management of Uninterruptible Power Supply (UPS) units and other power devices often deployed in small offices, and in IT installations subject to an erratic public power supply. The UPS units typically interface to an Attachment Daemon in the system they protect. This daemon is in turn polled by a Management Daemon which notifies users and system administrators of power supply incidents, and takes system shutdown decisions. The commands and responses described by this text are exchanged between the UPS Attachment Daemon and the Management Daemon. Current practice leads to weak security and this is addressed in the Security and IANA Considerations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 7 September 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. [Introduction](#)
 - 1.1. [How to read this text](#)
 - 1.2. [Current practice](#)
 - 1.3. [Requirements Language](#)
 - 1.4. [Comments](#)
2. [Terminology](#)
 - 2.1. [Attachment Daemon](#)
 - 2.2. [Driver](#)
 - 2.3. [Event](#)
 - 2.4. [Instant Command](#)
 - 2.5. [Management Daemon](#)
 - 2.6. [NUT \(Network UPS Tools\) Project](#)
 - 2.7. [Primary](#)
 - 2.8. [Secondary](#)
 - 2.9. [Session](#)
 - 2.10. [UPS status](#)
 - 2.11. [UPS variable](#)
3. [Protocol overview](#)
4. [Protocol specification](#)
 - 4.1. [Notation used in this specification](#)
 - 4.2. [Commands](#)
 - 4.2.1. [FSD](#)
 - 4.2.2. [GET](#)
 - 4.2.2.1. [CMDDESC](#)
 - 4.2.2.2. [DESC](#)
 - 4.2.2.3. [NUMLOGINS](#)
 - 4.2.2.4. [TYPE](#)
 - 4.2.2.5. [UPSDESC](#)
 - 4.2.2.6. [VAR](#)
 - 4.2.3. [HELP](#)
 - 4.2.4. [INSTCMD](#)
 - 4.2.5. [LIST](#)
 - 4.2.5.1. [CLIENT](#)
 - 4.2.5.2. [CMD](#)
 - 4.2.5.3. [ENUM](#)
 - 4.2.5.4. [RANGE](#)
 - 4.2.5.5. [RW](#)

- [4.2.5.6. UPS](#)
 - [4.2.5.7. VAR](#)
 - [4.2.6. LOGIN](#)
 - [4.2.7. LOGOUT](#)
 - [4.2.8. PASSWORD](#)
 - [4.2.9. PRIMARY](#)
 - [4.2.10. PROTVR](#)
 - [4.2.11. SET](#)
 - [4.2.12. STARTTLS](#)
 - [4.2.13. USERNAME](#)
 - [4.2.14. VER](#)
- [4.3. Error responses](#)
- [5. Statuses and Events](#)
 - [5.1. Status symbols](#)
 - [5.2. Events](#)
- [6. IANA Considerations](#)
 - [6.1. Namespaces used by Command, Responses, Statuses and Variables](#)
 - [6.2. Port name and number used to manage UPS units](#)
 - [6.2.1. Current situation](#)
 - [6.2.1.1. Port nut/3493](#)
 - [6.2.1.2. Port ups/401](#)
 - [6.2.2. Request to IANA](#)
- [7. Security Considerations](#)
 - [7.1. Agent verification](#)
 - [7.2. Encryption](#)
 - [7.3. Current security practice](#)
 - [7.4. Security needs](#)
 - [7.4.1. Attachment Daemon shim](#)
 - [7.4.2. Management Daemon shim](#)
- [8. Acknowledgments](#)
- [9. Normative References](#)
- [10. Informative References](#)
- [Appendix A. Variables and Instant Commands](#)
 - [A.1. UPS variables](#)
 - [A.2. UPS readable and writable variables](#)
 - [A.3. UPS Instant Commands](#)
- [Appendix B. Change Log](#)
 - [B.1. Changes in version 01](#)
 - [B.2. Changes in version 02](#)
- [Author's Address](#)

1. Introduction

1.1. How to read this text

The editor recommends that you read the HTML version of this text. It renders the protocol symbols such as OL correctly without quotation marks.

To lighten the text, the term "UPS" is used when "Managed Power Device" would be more complete. The reader should understand the simple "UPS" to include other managed power devices.

Ed: To be removed. The Idnits tool signals false positives when reading variable names such as ups.delay.shutdown. It wrongly assumes that they are domain names and applies RFC2606 which calls for use of domain names such as example.com

1.2. Current practice

This text documents UPS management techniques and current UPS management practice published by the [NUT Project \(2.6\)](#) which has been operational since 1998.

Since May 2002, the protocol described by this text has been operating on IANA port nut/3493 running over TCP.

Historically, the previous number 3305 was a relic of the original code's ancestry, and conflicted with other services. UDP support was dropped in July 2003. It had been deprecated for some time and was only capable of the simplest query commands as authentication is impossible over a UDP socket.

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119 \[RFC2119\]](#).

1.4. Comments

The editor welcomes comments. Technical matters should be addressed to the [NUT Project \(2.6\)](#)'s [mailing list \[mailinglist\]](#). Editorial matters may be addressed directly to the editor, email: ietf@rogerprice.org .

2. Terminology

The following technical terms appear in this text.

2.1. Attachment Daemon

The Attachment Daemon talks to the UPS units or other power devices often through a [Driver \(2.2\)](#) specific to the hardware model and the connection medium, e.g. USB, serial. It maintains an abstracted view of the hardware through the use of hardware statuses [2.10](#). A [Management Daemon \(2.5\)](#) may consult the abstracted view using the commands described in this text. Although the term "Attachment Daemon" includes the word "daemon" nothing requires that it be

implemented as a detached software service. An Attachment Daemon must support [statuses](#) (2.10) OB and OL. It must also support status LB if the UPS provides such information.

2.2. Driver

A Driver is that part of an Attachment Daemon which is specific to the hardware, the connection medium and the connection protocol, e.g. USB, serial. In current practice the Attachment Daemon has a driver for each hardware interface type it supports. Although this text considers the driver to be part of the Attachment Daemon, current practice is to see it as a separate software unit running as an unattached daemon "in front of" the Attachment Daemon. The protocol between the driver and the attachment daemon may use [RFC1628](#).

2.3. Event

An Event is a change in [UPS status](#) (2.10) detected by the [Management Daemon](#) (2.5)

2.4. Instant Command

A command which when sent to the [Attachment Daemon](#) (2.1) causes the hardware to immediately perform a function. For example INSTCMD su700 test.panel.start

2.5. Management Daemon

The Management Daemon is primarily responsible for managing the hardware and the system reaction to power loss. Using commands sent to the [Attachment Daemon](#) (2.1) it follows the status of the UPS and determines when UPS events occur. It takes decisions based on the events, such as calling for a system shutdown. Although the term includes the word "daemon" nothing requires that it be implemented as a detached software service. The Management Daemon may also provide administrative functions such as a graphic interface to view the hardware activity.

2.6. NUT (Network UPS Tools) Project

The primary goal of the [NUT project](#) [NUT] is to provide support for Power Devices, such as Uninterruptible Power Supplies. The Project has been in operation since 1998 with a major rework in 2003. It operates through a [mailing list](#) [mailinglist] and a [web site](#) [NUT].

2.7. Primary

When a power device such as a UPS unit supplies power to more than one system, the one to which the data lead is connected is known as

the primary. The others are secondary. See figure 4. Common current practice for system administrators is to consider the [Management Daemon](#) (2.5) in the primary to be the Primary Management Daemon which is in charge of the shutdown of all the systems powered by the UPS. The Primary Management Daemon sets status symbol FSD to order the secondaries to shut down.

Note: Historically, the primary was known as the "Master".

2.8. Secondary

When a hardware device such as a UPS unit supplies power to more than one system, the one to which the data lead is connected is known as the [Primary](#) (2.7). The other are secondaries. There is no [Attachment Daemon](#) (2.1) in a secondary. See figure 4. Common current practice for system administrators is to consider the [Management Daemon](#) (2.5) in a secondary to be a Secondary Management Daemon which understands status symbol FSD as an order to shut down.

Note: Historically, the secondary was known as the "Slave".

2.9. Session

The [Management Daemon](#) (2.5) may open a session with a specified device such as a UPS known to the [Attachment Daemon](#) (2.1). The session structure provides for audit and security as well as access to mission critical UPS functions. For example good practice requires a password protection for an [instant command](#) (2.4) which turns off a UPS outlet. Other than the commands and responses used, the details of session management are outside the scope of this text.

2.10. UPS status

The status of a hardware device such as a UPS unit is a symbolic description of the state of the unit. It consists of a space separated list of symbols from the set {ALARM BOOST BYPASS CAL CHRG COMM DISCHRG FSD LB NOCOMM OB OFF OVER RB RB TEST TRIM}. The symbols TICK and TOCK are experimental additions to the status and are not in common current practice. See [the appendix](#) (5.1) which specifies each of these symbols. The statuses LB, OB and OL are considered fundamental and are supported for all units. Other statuses depend on the feature set of the hardware.

2.11. UPS variable

The features provided by each UPS are represented by variables giving the current value attached to that feature. The UPS variable is an abstraction of the UPS hardware configuration and activity maintained by the [Attachment Daemon](#) (2.1). See [the appendix](#) ([Section](#)

[A.1, Paragraph 1](#)) which provides a non exhaustive list of variables. For example the variable `battery.charge` contains the current charge of the UPS battery as a percentage value.

3. Protocol overview

Figure [1](#) shows a reference configuration in which the command/response protocol applies. The UPS shown is representative of all power devices,

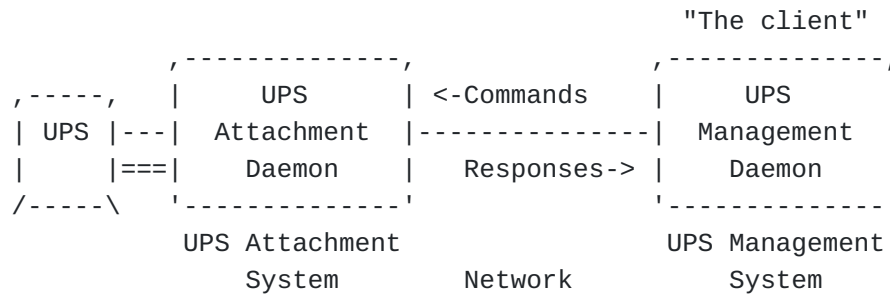


Figure 1: Reference Configuration

The reference configuration in figure [1](#) shows a single UPS unit which has a power supply link (===) and a data link (---) attached to a system running an [Attachment Daemon](#) ([2.1](#)). The UPS provides power supply protection to the system running the Attachment Daemon. The data link may use [RFC1628](#).

In practice there may be more than one UPS unit, and a unit may provide power protection to more than one system. The figure also shows a single [Management Daemon](#) ([2.5](#)). In practice there may be more than one Management Daemon, and any one Management Daemon may manage more than one UPS Attachment Daemon.

The protocol applies to connections between the Management Daemon and the Attachment daemon. The Management Daemon is known as the **client**. It sends commands over TCP to the Attachment Daemon and receives responses over TCP from that daemon.

The two daemons may run in the same system, or may be connected through a local or wide area network. In simple cases, as shown in figure [2](#), the [Attachment Daemon](#) ([2.1](#)) and the [Management Daemon](#) ([2.5](#)) are in the same system, the one protected by the UPS. The commands and responses are exchanged through an internal loopback interface.

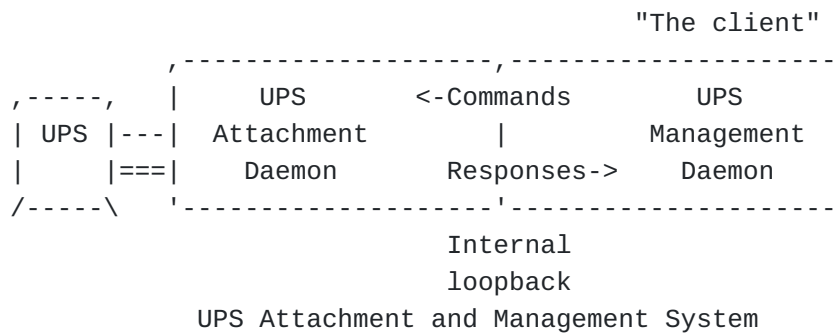


Figure 2: Simplified single-system configuration

The reference configuration does not require any specific design. For example figure 3 shows an arrangement in which the [Attachment Daemon](#) (2.1) is closely associated with, or even included in the UPS system setup. This is becoming more prevalent with the availability of low cost processors able to run the [Attachment Daemon](#) (2.1)

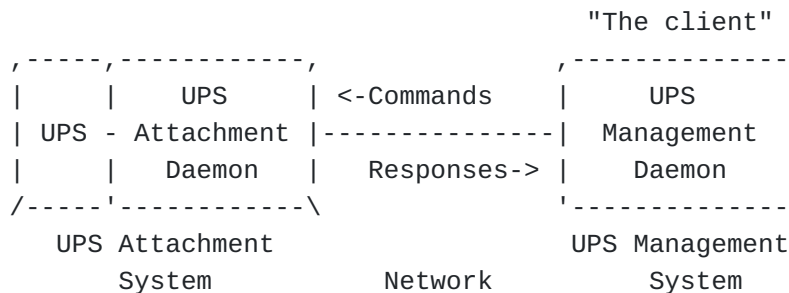


Figure 3: UPS and Attachment Daemon integration

As the power requirements for processors decrease, it is becoming increasingly common to use a single UPS to protect multiple systems as shown in figure 4. However there is only one data line (---) from the UPS to the [Primary](#) (2.7) system. The others have only power connections (===) to the UPS, and each is known as a [Secondary](#) (2.8). A [Secondary](#) (2.8) does not run an [Attachment Daemon](#) (2.1), it connects over a network to the [Attachment Daemon](#) (2.1) in the primary. Figure 4 shows the [Attachment Daemon](#) (2.1) and the primary [Management Daemon](#) (2.5) in the same system. This is common practice but it is not a technical requirement.

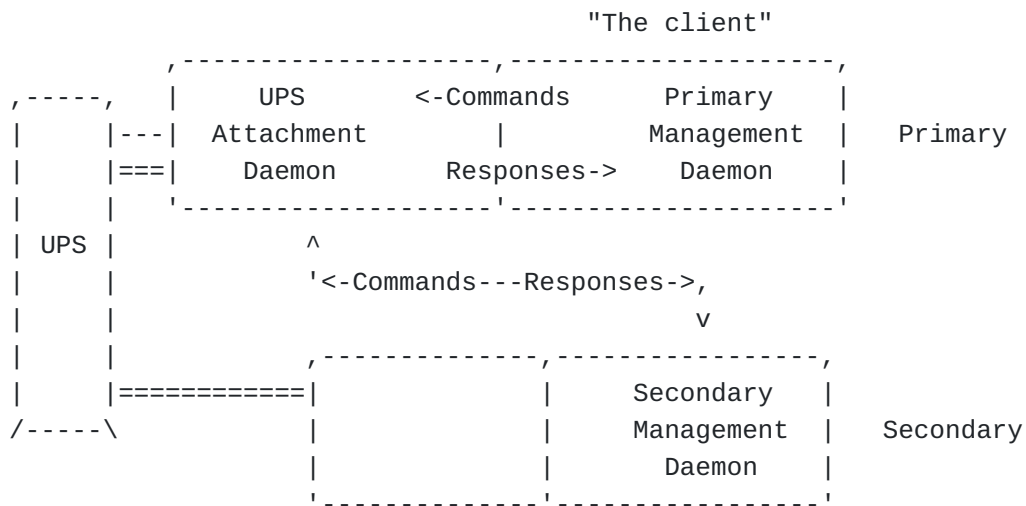


Figure 4: UPS protects multiple systems

4. Protocol specification

This specification includes only the commands and their responses. An implementation of the [Attachment Daemon \(2.1\)](#) has an internal state machine, and some complex implementations of the [Management Daemon \(2.5\)](#) include an internal state machine; for example to assist the system shutdown of a complex installation. However the management protocol used between them is effectively stateless. Most responses received by the [Management Daemon \(2.5\)](#) are sufficient in themselves, and at most require knowledge of the previous response to that command. See for example [table \(5.2\)](#) which maps the new ups.status response and the previous ups.status response to an [Event \(5.2\)](#) which is taken as the basis for [Management Daemon \(2.5\)](#) action.

4.1. Notation used in this specification

The character set used for commands and responses is UTF-8 but current practice is to limit the character set used to the single byte UTF-8 characters 0-127.

Multi-word elements are contained within U+0022 QUOTATION MARK characters for easier parsing. E.g. "UPS on fire". Embedded quotation marks are escaped with U+005C REVERSE SOLIDUS \ often known as backslashes. Embedded backslashes are also escaped by representing them as \.

Commands and responses have no leading or trailing whitespace, and are terminated with a single new line character U+000A LINE FEED (LF).

White space within commands and responses is reduced to one U+0020 (SP) SPACE.

4.2. Commands

4.2.1. FSD

A [Management Daemon \(2.5\)](#) which is [Primary \(2.7\)](#) and has the required authority, uses this command to set status symbol FSD in the [Attachment Daemon \(2.1\)](#). Current practice uses the symbol to tell each [Secondary \(2.8\)](#) system to shut down. It has the same value as the pair of status symbols OB LB.

Command: FSD <upsname> The response is: OK FSD-SET if the command is successful.

Current practice requires that an application introduce security controls in its session management to defend against abusive use of this command. The details are outside the scope of this text.

4.2.2. GET

Retrieve a single response from the server. The possible sub-commands are:

4.2.2.1. CMDDESC

Command: GET CMDDESC <upsname> <cmdname>

Response: CMDDESC <upsname> <cmdname> "<description>"

For example: GET CMDDESC su700 load.on and response CMDDESC su700 load.on "Turn on the load immediately"

This is like DESC, but it applies to an [instant command \(2.4\)](#).

4.2.2.2. DESC

Command: GET DESC <upsname> <varname>

Response: DESC <upsname> <varname> "<description>"

where <description> is a string that gives a brief explanation of the named variable. The [Attachment Daemon \(2.1\)](#) may return "Unavailable" if the file which provides this description is not installed.

For example command GET DESC su700 ups.status and response DESC su700 ups.status "UPS status"

4.2.2.3. NUMLOGINS

Command: GET NUMLOGINS <upsname>

Response: NUMLOGINS <upsname> <value>

where <value> is the number of clients which have succeeded in doing a LOGIN to this UPS.

For example command GET NUMLOGINS su700 and response NUMLOGINS su700 1

This information may be needed by the [Management Daemon \(2.5\)](#) to determine how many clients are still connected when starting the system shutdown process.

4.2.2.4. TYPE

Command: GET TYPE <upsname> <varname>

Response: TYPE <upsname> <varname> <type>...

where <type> can be one or more of the following tokens. Multiple types may be returned.

For example command GET TYPE su700 input.transfer.low and response TYPE su700 input.transfer.low ENUM

Type	Meaning
RW	This variable may be set to another value with command SET
ENUM	An enumerated type, which supports specific pre-determined values
STRING:n	This is a string of maximum length n
RANGE	This is a number, either integer or float, comprised in the range which may be seen with the command LIST RANGE (4.2.5.4)
NUMBER	This is a single numeric value, either integer or float

Table 1: Variable types

Notes:

*ENUM, STRING:n and RANGE, are usually associated with RW, but not always. The default <type>, when omitted, is numeric, so either integer or float. Each [Driver \(2.2\)](#) is then responsible for handling values as either integer or float.

*Current practice is to represent floating point values using locale C.utf8 which is a decimal (base 10) US English-based

representation. Hexadecimal, exponents, and comma for thousands separator are not allowed. For example: "1200.20" is valid, while "1,200.20" and "1200,20" are invalid.

4.2.2.5. UPSDESC

Command: GET UPSDESC <upsname>

Response: UPSDESC <upsname> "<description>"

where <description> is defined by the [Attachment Daemon \(2.1\)](#) configuration. If it is not set, current practice is for the [Attachment Daemon \(2.1\)](#) to return "Unavailable".

For example command GET UPSDESC su700 and response UPSDESC su700 "Development box"

This can be used to provide human-readable descriptions instead of a cryptic upsname@hostname string.

4.2.2.6. VAR

Command: GET VAR <upsname> <varname>

Response: VAR <upsname> <varname> "<value>"

For example command GET VAR su700 ups.status and response VAR su700 ups.status "OB LB"

4.2.3. HELP

Return a list of the commands supported by the [Attachment Daemon \(2.1\)](#). This command is intended for human as well as program use.

Command HELP

For example, the following command line sequence executed on an [Attachment Daemon \(2.1\)](#):

```
netcat localhost 3493
```

```
HELP
```

```
Commands: HELP VER GET LIST SET INSTCMD LOGIN LOGOUT  
          LOGOUT USERNAME PASSWORD STARTTLS
```

4.2.4. INSTCMD

Send an [instant command \(2.4\)](#) to the UPS.

Command: INSTCMD <upsname> <cmdname> The response is: OK

where <upsname> is the name of the UPS and <cmdname> is the [instant command](#) (2.4) to be issued to that UPS.

For example the command: INSTCMD su700 test.panel.start and the response OK

4.2.5. LIST

The LIST commands all produce a response with a common container format. The response will begin with BEGIN LIST and then repeat the initial query. A list then follows, with as many lines as are necessary. The response ends with END LIST followed by the initial query.

The formatting may seem a bit redundant, but it makes a different form of client possible. A client can send a LIST query and then go off and wait for the response. When it arrives, the [Management Daemon](#) (2.5) doesn't need a complicated state machine to remember which list is which.

The possible subcommands are:

4.2.5.1. CLIENT

The command calls for the [Attachment Daemon](#) (2.1) to report all the current [Management Daemon](#) (2.5) clients of a given UPS. See [command LOGIN](#) (4.2.6).

Command: LIST CLIENT <upsname> The response is

```
BEGIN LIST CLIENT <upsname>
CLIENT <upsname> <client_IP_address>
...
END LIST CLIENT <upsname>
```

For example, the command LIST CLIENT ups1 and the response:

```
BEGIN LIST CLIENT ups1
CLIENT ups1 ::1
CLIENT ups1 198.51.100.2
END LIST CLIENT ups1
```

4.2.5.2. CMD

The command calls for the [Attachment Daemon](#) (2.1) to report a list of each UPS [instant command](#) (2.4) which the [Management Daemon](#) (2.5) may send to the [Attachment Daemon](#) (2.1). This [instant command](#) (2.4)

list is the abstracted view of the UPS hardware capabilities. An economical UPS will support few or no [instant command](#) (2.4) but a professional model should support more.

Command: LIST CMD <upsname> The response is:

```
BEGIN LIST CMD <upsname>
CMD <upsname> <cmdname>
...
END LIST CMD <cmdname>
```

where <upsname> is the name of the UPS, and <cmdname> is the name of the command which may be issued to the UPS.

For example the command: LIST CMD su700 and the response:

```
BEGIN LIST CMD su700
CMD su700 load.on
CMD su700 test.panel.start
...
END LIST CMD su700
```

4.2.5.3. ENUM

The command calls for the [Attachment Daemon](#) (2.1) to report the set of possible values of a UPS variable which has predetermined values.

Command: LIST ENUM <upsname> <varname> The response is:

```
BEGIN LIST ENUM <upsname> <varname>
ENUM <upsname> <varname> "<value>"
...
END LIST ENUM <upsname> <varname>
```

where <upsname> is the name of the UPS, <varname> is the UPS variable and <value> is one of the possible values of the UPS variable. Note that the U+0022 QUOTATION MARK characters are part of the response.

For example the command: LIST ENUM su700 input.transfer.low and the response:

```
BEGIN LIST ENUM su700 input.transfer.low
ENUM su700 input.transfer.low "103"
ENUM su700 input.transfer.low "100"
...
END LIST ENUM su700 input.transfer.low
```

4.2.5.4. RANGE

The command calls for the [Attachment Daemon \(2.1\)](#) to report the interval in which valid values of a UPS variable lie.

Command: LIST RANGE <upsname> <varname> The response is:

```
BEGIN LIST RANGE <upsname> <varname>
RANGE <upsname> <varname> "<min>" "<max>"
...
END LIST RANGE <upsname> <varname>
```

where <upsname> is the name of the UPS, <varname> is the UPS variable and {<min>,<max>} is the interval of valid values of the UPS variable. Note that the U+0022 QUOTATION MARK characters are part of the response.

For example, the command LIST RANGE su700 input.transfer.low and the response:

```
BEGIN LIST RANGE su700 input.transfer.low
RANGE su700 input.transfer.low "90" "100"
RANGE su700 input.transfer.low "102" "105"
...
END LIST RANGE su700 input.transfer.low
```

4.2.5.5. RW

The command calls for the [Attachment Daemon \(2.1\)](#) to report a list of the UPS variables associated with a given UPS which may be read and written by the [Management Daemon \(2.5\)](#). These variables are the abstracted view of the UPS hardware capabilities. An economical UPS will support few variables but a professional model should support at least the variables which are needed for automatic shutdown and restart.

Command: LIST RW <upsname>

The response is:

```
BEGIN LIST RW <upsname>
RW <upsname> <varname> "<value>"
...
END LIST RW <upsname>
```

where <upsname> is the name of the UPS, <varname> is the UPS variable and <value> is the value of the UPS variable. Note that the U+0022 QUOTATION MARK characters are part of the response.

For example the command: LIST RW su700 and the response:

```
BEGIN LIST RW su700
RW su700 output.voltage.nominal "115"
RW su700 ups.delay.shutdown "020"
...
END LIST RW su700
```

4.2.5.6. UPS

Command: LIST UPS

The response is:

```
BEGIN LIST UPS
UPS <upsname> "<description>"
...
END LIST UPS
```

where <upsname> is the name of a UPS, and <description> is the value of the description maintained by the [Attachment Daemon \(2.1\)](#) if available. It is set to "Unavailable" otherwise. Note that the U+0022 QUOTATION MARK characters are part of the response.

This command can also be used to determine what values of <upsname> are valid before calling other functions on the server. This is also a good way to handle situations where a single [Attachment Daemon \(2.1\)](#) supports multiple UPS's. It is also useful for clients which perform a UPS discovery process.

For example, the response:


```
BEGIN LIST UPS
UPS su700 "Development box"
END LIST UPS
```

4.2.5.7. VAR

Command: LIST VAR <upsname> The response is:

```
BEGIN LIST VAR <upsname>
VAR <upsname> <varname> "<value>"
...
END LIST VAR <upsname>
```

where <upsname> is the name of the UPS, <varname> is the UPS variable and <value> is the value of the UPS variable. Note that the U+0022 QUOTATION MARK characters are part of the response.

The response to this command lists the UPS variables available for this UPS and their current values. For example the command LIST VAR su700 and the response:

```
BEGIN LIST VAR su700
VAR su700 ups.mfr "APC"
VAR su700 ups.mfr.date "10/17/96"
...
END LIST VAR su700
```

4.2.6. LOGIN

The [Attachment Daemon \(2.1\)](#) provides facilities to limit access to the UPS unit(s) to which it is attached. A system administrator performs a login to open a [Session \(2.9\)](#) and gain access to a UPS, and a logout when the [Session \(2.9\)](#) is no longer needed.

Command: LOGIN <upsname> The response is OK if the login is successful.

Current practice is for the command LOGIN to allow connection to the [Attachment Daemon \(2.1\)](#). If there are two UPS units attached to the [Attachment Daemon \(2.1\)](#) logging into the first will automatically log into the second as well.

4.2.7. LOGOUT

The [Attachment Daemon \(2.1\)](#) provides facilities to limit access to the UPS unit(s) to which it is attached. A system administrator performs a login to open a [Session \(2.9\)](#) and gain access, and a logout when the [Session \(2.9\)](#) is no longer needed. The command LOGOUT allows the system administrator to disconnect gracefully.

Command: LOGOUT with the response OK Goodbye if the disconnection was successful.

4.2.8. PASSWORD

The [Management Daemon \(2.5\)](#) specifies a password required to enter a [Session \(2.9\)](#) with the [Attachment Daemon \(2.1\)](#).

Command: PASSWORD <password> The response is: OK if the command is successful.

4.2.9. PRIMARY

The [Attachment Daemon \(2.1\)](#) uses this command within a [Session \(2.9\)](#) to claim that it is a [Primary \(2.7\)](#) and has the required authority to perform such critical actions as setting status symbol FSD.

Command: PRIMARY <upsname> with response: OK if the [Attachment Daemon \(2.1\)](#) does have such authority.

where <upsname> is the name of the UPS.

Note: Historically, this command was known as MASTER.

4.2.10. PROTVER

Return the implementation version of the command/response protocol used by the [Attachment Daemon \(2.1\)](#). This command is intended for human as well as program use.

Command PROTVER

For example, the following command line sequence in the [Attachment Daemon \(2.1\)](#):

```
netcat localhost 3493
PROTVER
1.2
```

Note: Historically, this command was known as NETVER and current practice is to use NETVER instead of PROTVER

4.2.11. SET

The command calls for the [Attachment Daemon \(2.1\)](#) to set a UPS variable to a given value. Whether this has an effect on the UPS hardware is specific to the [Driver \(2.2\)](#) and the UPS model.

Command: SET VAR <upsname> <varname> "<value>" The
response is: OK

where <upsname> is the name of the UPS, <varname> is the UPS variable and <value> is the value to be assigned to the UPS variable. Note that the U+0022 QUOTATION MARK characters are part of the command.

For example the command: SET VAR su700 ups.id "My UPS" and the response OK

4.2.12. STARTTLS

The client tells the [Attachment Daemon \(2.1\)](#) to switch to TLS encrypted communication. When the client receives OK it also switches to TLS encryption.

Command: STARTTLS The response is: OK STARTTLS if the
command is successful.

4.2.13. USERNAME

The [Session \(2.9\)](#) may require that the [Management Daemon \(2.5\)](#) identifies a "UPS management user" who has been authorized to send commands to the [Attachment Daemon \(2.1\)](#). In current practice, this "user" is specific to UPS management and is not a user of the operating system. The technique for specifying this UPS management user is specific to the implementation, and may be done with a configuration file.

Command: USERNAME <username> The response is OK if the
command is successful.

4.2.14. VER

Return the implementation version of the [Attachment Daemon \(2.1\)](#). This command is intended for human as well as program use.

Command VER

For example, the following command line sequence:

```
netcat localhost 3493
```

```
VER
```

```
Network UPS Tools upsd 2.7.4 - http://www.networkupstools.org/
```

4.3. Error responses

Error responses have the following format:

```
ERR <error-name> [<extra>]
```

where <error-name> is a single word token taken from the 27 characters A-Z and HYPHEN-MINUS U+002D. Implementations may if needed add an additional optional <extra>. Current practice does not make use of this possibility.

The <error-name> may have one of the following values:

The error name token <error-name>	Meaning
ACCESS-DENIED	The client's host and/or authentication details (username, password) are not sufficient to execute the requested command.
ALREADY-LOGGED-IN	The client has already sent a successful LOGIN command for a given UPS and can't do it again. Note: Current practice is to impose a limit of one LOGIN record per connection. If there are two UPS's attached to the Attachment Daemon (2.1) , logging into one of them automatically logs into the other as well. See command LOGIN (4.2.6) .
ALREADY-SET-PASSWORD	The client has already supplied a PASSWORD and is attempting to repeat the command in the same Session (2.9) .
ALREADY-SET-USERNAME	The client has already supplied a USERNAME, and is attempting to repeat the command within the same Session (2.9) .
ALREADY-SSL-MODE	

The error name token <error-name>	Meaning
	TLS/SSL mode is already enabled on this connection, so the Attachment Daemon (2.1) can't start it again.
CMD-NOT-SUPPORTED	The specified UPS doesn't support the instant command (2.4) command.
DATA-STALE	The Attachment Daemon (2.1) is connected to the Driver (2.2) for the UPS, but that driver isn't providing regular updates or has specifically marked the data as stale. Current practice is for the Attachment Daemon (2.1) to refuse to provide the Management Daemon (2.5) with variables on stale units to avoid false readings. This generally means that the Driver (2.2) is running, but it has lost communication with the hardware. Check the physical connection to the equipment.
DRIVER-NOT-CONNECTED	The Attachment Daemon (2.1) can't perform the requested command, since the Driver (2.2) for that UPS is not connected. This usually means that the driver is not running, or if it is, is misconfigured.
FEATURE-NOT-CONFIGURED	This instance of the Attachment Daemon (2.1) hasn't been configured properly to allow the requested feature to operate. In current practice this error response is possible only for command STARTTLS.
FEATURE-NOT-SUPPORTED	This instance of Attachment Daemon (2.1) does not support the requested feature. In current practice this error response is possible only for command STARTTLS.
INSTCMD-FAILED	The Attachment Daemon (2.1) failed to deliver the instant command (2.4) request to the Driver (2.2) . No further information is available to the client. This typically indicates a dead or broken driver.
INVALID-ARGUMENT	The client sent an argument to a command which is not recognized or is otherwise invalid in this context. This is typically caused by sending a valid

The error name token <error-name>	Meaning
	command such as GET with an invalid subcommand.
INVALID-PASSWORD	The client sent an invalid password.
INVALID-USERNAME	The client sent an invalid username.
INVALID-VALUE	The value specified in the request is not valid. This usually applies to a SET of an ENUM type which is using a value not in the list of allowed values.
PASSWORD-REQUIRED	The command requires a password for authentication, but the client hasn't provided one.
READONLY	The requested variable in a SET command is not writable.
SET-FAILED	The Attachment Daemon (2.1) failed to deliver the set request to the Driver (2.2) . This is similar to INSTCMD-FAILED.
TOO-LONG	The requested value in a SET command is too long.
UNKNOWN-COMMAND	The Attachment Daemon (2.1) doesn't recognize the command.
UNKNOWN-UPS	The UPS specified in the request is not known to the Attachment Daemon (2.1) . This usually means that it didn't match anything in the Attachment Daemon (2.1) configuration.
USERNAME-REQUIRED	The command requires a username for authentication, but the client hasn't provided one.
VAR-NOT-SUPPORTED	The specified UPS doesn't support the UPS variable (2.11) in the command.

Table 2: Error responses

5. Statuses and Events

5.1. Status symbols

These symbols resume the abstracted view of the UPS hardware maintained by the [Attachment Daemon \(2.1\)](#). The variable `ups.status` contains one or more space-separated status symbols which together describe the UPS state at that instant. In current practice the [Management Daemon \(2.5\)](#) will poll variable `ups.status` every 5 seconds with a command such as `GET VAR su700 ups.status` and response `VAR su700 ups.status "OB LB"` to discover changes in the UPS status. These changes will indicate UPS events.

Symbol	Meaning
ALARM	The UPS reports that it requires intervention.
BOOST	The UPS has determined the voltage level of the public supply is too low, and is boosting it to the required level. The UPS continues to supply the protected system from the public supply.
BYPASS	The UPS is feeding current directly from the public supply to the protected system. The backup facilities are disconnected. This state allows maintenance personnel to change the batteries without interrupting the protected system.
CAL	The UPS is calibrating itself, for example to determine at what charge the LB status is raised or lowered.
CHRG	The UPS battery is charging. This usually implies that the UPS also has status OL, but may not be the case if the UPS also has status OFF. Note: OL does not imply CHRG if the battery is floating.
COMM	The Attachment Daemon (2.1) has effective contact with the UPS.
DISCHRG	The UPS battery is discharging. This usually implies that the UPS also has status OB, but may not be the case if the UPS also has status OFF. Note: OB does not imply DISCHRG if the battery is floating.
FSD	This "Forced Shut Down" status signals that the final shutdown sequence has begun.
LB	Low Battery. The battery level of the UPS is below a chosen limit. The UPS may be in status OL or OB.
NOCOMM	The Attachment Daemon (2.1) has no effective contact with the UPS.
OB	On Battery. The UPS is offline, taking energy from it's battery. The battery is discharging. A UPS must have status OB or OL, otherwise it is deemed dead.
OFF	The UPS is in state "Off". It does not react to failure in the public power supply. The exact meaning depends on the model.
OL	OnLine. The UPS is online, receiving energy from the public supply. The battery is charging. A UPS must have status OB or OL, otherwise it is deemed dead.
OVER	Overloaded. The UPS reports that the load on it is beyond it's normal operating maximum.
RB	Replace battery. The UPS reports that it's battery/batteries should be replaced.
TEST	Under test. The UPS is currently undergoing a test, which may have been called for manually or internally.
TICK	Heartbeat. A software UPS in the Attachment Daemon (2.1) provides a regular signal monitored by the Management Daemon

Symbol	Meaning
	(2.5) as a way of verifying effective end-to-end management. TICK and TOCK are companions.
TOCK	Heartbeat. See TICK
TRIM	The UPS has determined that the voltage level of the public supply is too high, and is reducing it to the required level. The UPS continues to supply the protected system from the public supply.

Table 3: UPS status symbols

5.2. Events

A [Management Daemon](#) (2.5) deduces the occurrence of a UPS Event from a change in the [UPS status](#) (2.10) received from the [Attachment Daemon](#) (2.1). The following table summarizes the process. A status of "none" means that the status symbol is not present in the variable ups.status. In current practice, the variable ups.status is retrieved every 5 seconds. The "old" status is therefore the previous value retrieved 5 seconds ago.

Current practice is for the [Management Daemon](#) (2.5) to assign names to certain events. These is shown in the table in parentheses.

Old status	New status	Event		Old status	New status	Event
none	ALARM	Alarm on		ALARM	none	Alarm off
none	BOOST	Boosting voltage		BOOST	none	Not boosting
none	BYPASS	Bypass on		BYPASS	none	Bypass off
none	CAL	Calibrating		CAL	none	Not calibrating
none	CHRG	Charging		CHRG	none	Not charging
none	COMM	UPS communicating (COMMOK)		COMM	none	[Note 5]
none	DISCHRG	Discharging		DISCHRG	none	Not discharging
none	FSD	System shutdown (FSD) (SHUTDOWN)		FSD	none	Shutdown abandoned [Note 1]
none	LB	Low battery [Note 2] (LOWBATT)		LB	none	Battery not low
none	NOCOMM	UPS dead? [Note 5]		NOCOMM	none	[Note 5]

Old status	New status	Event		Old status	New status	Event
		(COMMBAD) (NOCOMM)				
none	OFF	UPS turned off		OFF	none	UPS not turned off
OB	OL	Receiving wall power (ONLINE)		OL	OB	Wall power lost (ONBATT)
none	OVER	UPS overloaded		OVER	none	Overload gone
none	RB	Replace battery (REPLBATT)		RB	none	Replacement canceled
none	TEST	Test starts		TEST	none	Test finished
none	TICK	Heartbeat event [Note 4]		TICK	none	No heartbeat [Note 4]
none	TOCK	Heartbeat event [Note 4]		TOCK	none	No heartbeat [Note 4]
none	TRIM	Trimming voltage		TRIM	none	Not trimming

Table 4: Event deduction from status changes

Notes

1. Current practice does not include this event.
2. If the status OB is present, current practice takes [Management Daemon](#) (2.5) reception of LB as an order to perform an emergency system shutdown.
3. (For future use)
4. The use of a software defined UPS to provide a heartbeat is experimental and is not part of common current practice.
5. Current practice is: if the UPS has not responded for 15 seconds, the [Management Daemon](#) (2.5) assumes that the UPS is "dead" (NOCOMM), and if the last known OL/OB status was OB a system shutdown (FSD) is called for.

6. IANA Considerations

Ed: See [Guidelines for Writing an IANA Considerations Section in RFCs](#) [RFC8126].

This text raises five matters which fall within IANA Considerations:

1. The namespaces occupied by the [protocol commands](#) (4.2) described in this text.

2. The namespaces occupied by the [protocol responses](#) (4.3) described in this text.
3. The namespace occupied by [UPS status](#) (2.10) names,
4. The namespace occupied by [UPS variable](#) (2.11) names,
5. The port name and port number used to manage UPS units.

6.1. Namespaces used by Command, Responses, Statuses and Variables

Current [NUT Project](#) (2.6) experience after more than 20 years is that the UPS management area advances slowly, and that there are few requests to modify or extend the Commands, Responses, Statuses and Variables. When this does occur, the [NUT Project](#) (2.6) has been able to settle the matter without difficulty in the project mailing list. It is therefore proposed to not burden IANA with this namespace management and to continue with the current process in which the project in its mailing list acts as a Working Group,

6.2. Port name and number used to manage UPS units

6.2.1. Current situation

See the IANA [[Registry](#)] for the latest situation.

6.2.1.1. Port nut/3493

In 2002 IANA assigned port nut/3493 to project lead Russell Kroll, and updated the assignment to the [NUT Project](#) (2.6) itself in 2020.

6.2.1.2. Port ups/401

In 2008 IANA assigned ups/401 "Uninterruptible Power Supply" to Mr. Charles Bennett as both assignee and contact. We have been unable to find any protocol document or other published activity report for this port. Mr. Bennett himself died in 2015, see [obituary](#) [[Bennett](#)]. Since his email address was registered by IANA as bennettc@ohio.edu it is possible that the University of Ohio is a successor in interest. The editor tried to contact the IT support department of the university by email and telephone but was rejected. Ed: My non-contact was Mr. Keith Brock, IT Support Senior Specialist, brock@ohio.edu +1 740 597 2136

6.2.2. Request to IANA

The [NUT Project](#) (2.6) is mindful of IANA's position

that the applicant MUST NOT use the requested port in implementations deployed for use on the public Internet prior to the completion of the assignment

in [clause 8.1.1 of \[RFC6335\]](#) and makes the following requests to assist the security migration strategy described in [Section 7.4, Paragraph 1](#).

1. The [NUT Project \(2.6\)](#) wishes to use the existing UPS Management Protocol as described in this text on existing port ups/401 for full security access to the [Attachment Daemon \(2.1\)](#), i.e. command STARTTLS must be issued, accepted and applied.
2. The [NUT Project \(2.6\)](#) requests that IANA consider transferring port ups/401 to the [NUT Project \(2.6\)](#) and would appreciate IANA Expert advice on this matter.

7. Security Considerations

A functioning power supply is vital to a computing system. The [Management Daemon \(2.5\)](#) is able to shut down a working system and it's power supply: this raises multiple security issues. Most of these are well known IT issues concerning system protection and disaster recovery, and are beyond the scope of this text. However the protocol itself has security considerations:

1. It should not be possible for non-authorized agents to open sessions and send mission-critical commands such as FSD <upsname> to the [Attachment Daemon \(2.1\)](#).
2. It should not be possible to intercept the traffic between the [Attachment Daemon \(2.1\)](#) and the [Management Daemon \(2.5\)](#).

The following facilities address these requirements.

7.1. Agent verification

The protocol provides commands LOGIN and PASSWORD which allow a [Management Daemon \(2.5\)](#) to authenticate itself to the [Attachment Daemon \(2.1\)](#). The password needs protection from sniffing: This is done by encrypting the traffic.

7.2. Encryption

The protocol provides command STARTTLS which calls on the [Attachment Daemon \(2.1\)](#) to support TLS encryption of the communication. If this command is accepted, the [Management Daemon \(2.5\)](#) must also encrypt.

At present the command STARTTLS is too frequently refused, and traffic proceeds unencrypted, with for example plain text transmission of passwords.

7.3. Current security practice

Experience over the last 20 years shows that new UPS management software releases are not frequent, and when installed, stay unmodified for some years. This is probably because UPS management is a mature hardware dependent activity. A limited number of system administrators have access to the UPS hardware and software and tend to assume a certain "security by obscurity" since many installations have a configuration as shown in figure 5 which uses port nut/3493 between the two daemons running in the same system. The traffic is often not encrypted, and when encrypted uses deprecated early versions of SSL/TLS.

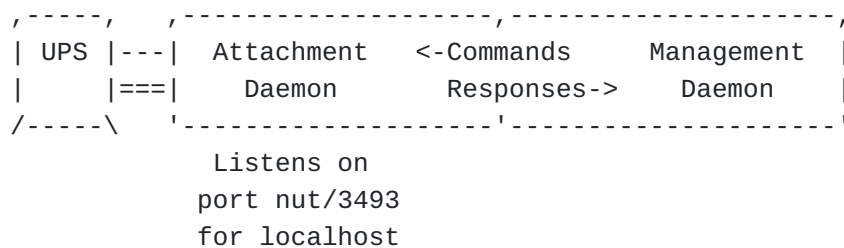


Figure 5: Common single-system configuration

This situation is now changing as low cost processors become available, costing significantly less than a UPS unit. This evolution makes it interesting to shift to a configuration as shown in figure 6, but it also exacerbates the security weakness of figure 5 since the traffic between the daemons is now over an exposed network.

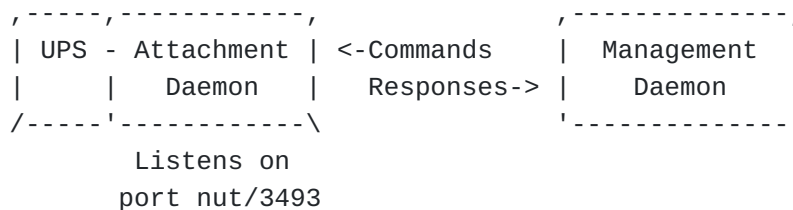


Figure 6: Integration of UPS and Attachment Daemon

7.4. Security needs

UPS management needs to move to a more secure practice in which all traffic is encrypted, but this cannot be imposed by a wave of the hand. The ideal would be an easy-to-follow migration plan which provides the required encryption but tolerates the slow moving updates of the UPS software. A possible technique introduces shims between the [Attachment Daemon \(2.1\)](#) and the network, and between the network and the [Management Daemon \(2.5\)](#) as shown in figure 7. These

shims provide TLS support, allowing the [Attachment Daemon \(2.1\)](#) and [Management Daemon \(2.5\)](#) to continue temporarily without native TLS. The technique has been successfully tested, but the principal difficulty is that the shims make use of a second port which is not currently available.

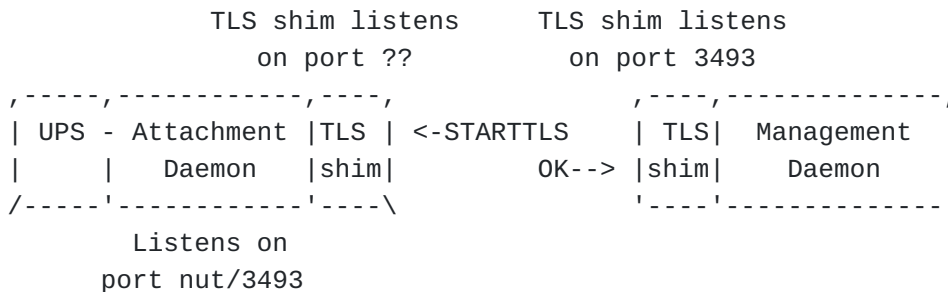


Figure 7: Shims provide TLS support during migration

7.4.1. Attachment Daemon shim

The shim in front of the [Attachment Daemon \(2.1\)](#) listens to incoming traffic on a port to be specified. When it receives the command STARTTLS it

1. Returns OK to the client and sets up TLS encapsulation.
2. Does not send STARTTLS to the [Attachment Daemon \(2.1\)](#) port nut/3493.

All other commands and responses are passed through.

7.4.2. Management Daemon shim

The shim in front of the [Management Daemon \(2.5\)](#) listens for incoming traffic on port nut/3493. When it receives the command STARTTLS it

1. Returns FEATURE-NOT-CONFIGURED to the client.
2. Sends STARTTLS to the [Attachment Daemon \(2.1\)](#) on a port to be specified.

All other commands and responses are passed through.

8. Acknowledgments

This text is based on the [NUT Project \(2.6\) documentation \[developerpage\]](#). The editor acknowledges the work of Charles Lepple, Arjen de Korte, Arnaud Quette, Jim Klimov, Russell Kroll, and many others who contribute to the nut-upsuser [mailing list \[mailinglist\]](#).

The source for the text is marked up using an [SGML DTD](#) [[SGML](#)] and an XML meta-DTD as defined by [HyTime Annex A](#) [[HyTimeA](#)]. The [sgmlnorm](#) [[sgmlnorm](#)] program generates XML which program [xml2rfc](#) [[RFC7991](#)] uses to prepare the text. The editor acknowledges the help received from Carsten Bormann and Julian Reschke in the xml2rfc mailing list.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

10. Informative References

[Bennett] "Charles Bennett Obituary", Publisher: Jagers and Sons Funeral Home, 24 Morris Ave., Athens OH, <<https://www.legacy.com/obituaries/name/charles-bennett-obituary?pid=174356861>>.

[developerguide] "Network UPS Tools (NUT) Project Developer Guide", <<https://networkupstools.org/docs/developer-guide.chunked/ar01s09.html>>.

[HyTimeA] "International Standard ISO/IEC 10744 -- Hypermedia/Time-based Structuring Language, Annex A, SGML Extended Facilities", ISO/IEC JTC 1/SC 34 Document description and processing languages, 1997.

[mailinglist] "Network UPS Tools (NUT) Project Mailing List", <<https://alioth-lists.debian.net/cgi-bin/mailman/listinfo/nut-upsuser>>.

[NUT] "Network UPS Tools (NUT) Project", <<https://www.networkupstools.org>>.

[Registry] "Service Name and Transport Protocol Port Number Registry", Publisher: IANA, <<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>>.

[RFC1628] Case, J., Ed., "UPS Management Information Base", RFC 1628, DOI 10.17487/RFC1628, May 1994, <<https://www.rfc-editor.org/info/rfc1628>>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC

6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, December 2016, <<https://www.rfc-editor.org/info/rfc7991>>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

[SGML] Goldfarb, Charles F., "The SGML Handbook", ISBN 0-19-853737-9, 1990.

[sgmlnorm] Clark, James., "SGMLNORM An SGML System Conforming to International Standard ISO 8879 -- Standard Generalized Markup Language", <<http://www.jclark.com/sp/sgmlnorm.htm>>.

Appendix A. Variables and Instant Commands

A.1. UPS variables

The UPS variables represent the abstracted state of the UPS unit. The number of variables depends on the sophistication of the product. Those shown here are a typical example. An implementation of a [Management Daemon \(2.5\)](#) acting as a utility program may provide a listing of the variables available, as well as acting on them. Ed:
NUT: upsc

Variable	Typical value	Default description
battery.charge	100	"Battery charge (percent of full)"
battery.charge.low	20	"Remaining battery level when UPS switches to LB (percent)"
battery.runtime	1481	"Battery runtime (seconds)"
battery.type	PbAc	"Battery chemistry"
device.mfr	WhizBang	""
device.model	Economy 1700	""
device.serial	1234567890	""
device.type	ups	""
driver.name	usbhid-ups	"Driver name"
driver.parameter.lowbatt	37	

Variable	Typical value	Default description
		"Driver parameter: <name>"
driver.parameter.offdelay	30	"Driver parameter: <name>"
driver.parameter.ondelay	40	"Driver parameter: <name>"
driver.parameter.pollfreq	30	"Driver parameter: <name>"
driver.parameter.pollinterval	2	"Driver parameter: <name>"
driver.parameter.port	auto	"Driver parameter: <name>"
driver.parameter.synchronous	no	"Driver parameter: <name>"
driver.parameter.vendorid	0999	"Driver parameter: <name>"
driver.version	2.7.4	"Driver version - NUT release"
driver.version.data	HID 1.39	""
driver.version.internal	0.41	"Internal driver version"
input.transfer.high	264	"High voltage transfer point (V)"
input.transfer.low	184	"Low voltage transfer point (V)"
outlet.1.desc	PowerShare Outlet 1	"Outlet description"
outlet.1.id	2	"Outlet system identifier"
outlet.1.status	on	"Outlet switch status"
outlet.1.switchable	no	"Outlet switch ability"
outlet.2.desc	PowerShare Outlet 2	"Outlet description"
outlet.2.id	3	"Outlet system identifier"
outlet.2.status	on	"Outlet switch status"
outlet.2.switchable	no	"Outlet switch ability"
outlet.desc	Main Outlet	"Outlet description"
outlet.id	1	"Outlet system identifier"
outlet.power	25	""
outlet.switchable	no	"Outlet switch ability"
output.frequency.nominal	50	"Nominal output frequency (Hz)"
output.voltage	230.0	"Output voltage (V)"

Variable	Typical value	Default description
output.voltage.nominal	230	"Nominal output voltage (V)"
ups.beeper.status	enabled	"UPS beeper status"
ups.delay.shutdown	20	"Interval to wait after shutdown with delay command (seconds)"
ups.delay.start	30	"Interval to wait before (re)starting the load (seconds)"
ups.firmware	02	"UPS firmware"
ups.load	20	"Load on UPS (percent of full)"
ups.mfr	WhizBang	"UPS manufacturer"
ups.model	Economy 1700	"UPS model"
ups.power.nominal	1700	"UPS power rating (VA)"
ups.productid	ffff	"Product ID for USB devices"
ups.serial	0000000000	"UPS serial number"
ups.status	OL	"UPS status"
ups.timer.shutdown	0	"Time before the load will be shutdown (seconds)"
ups.timer.start	0	"Time before the load will be started (seconds)"
ups.vendorid	0999	"Vendor ID for USB devices"

Table 5: Typical UPS Variables

A.2. UPS readable and writable variables

Some of the features of a UPS are represented by variables which may be tuned by the user. The following variables are typical of such tunable features. The precise list depends on the model of UPS. An implementation of a [Management Daemon \(2.5\)](#) acting as a utility program may provide a listing of the variables available, as well as acting on them. Ed: NUT: upsrw

Variable	Typical value	Default description provided as response to the command GET DESC
battery.charge.low	20	"Remaining battery level when UPS switches to LB (percent)"
input.transfer.high	264	"High voltage transfer point (V)"

Variable	Typical value	Default description provided as response to the command GET DESC
input.transfer.low	184	"Low voltage transfer point (V)"
outlet.1.desc	PowerShare Outlet 1	"Outlet description"
outlet.2.desc	PowerShare Outlet 2	"Outlet description"
outlet.2.switchable	no	"Outlet switch ability"
outlet.desc	Main Outlet	"Outlet description"
outlet.power	25	"Description unavailable"
output.voltage.nominal	230	"Nominal output voltage (V)"
ups.delay.shutdown	20	"Interval to wait after shutdown with delay command (seconds)"
ups.delay.start	30	"Interval to wait before (re)starting the load (seconds)"

Table 6: Typical readable and writable UPS Variables

A.3. UPS Instant Commands

Some of the features of a UPS are actions which may be ordered by the user. The following variables represent such actions. The precise list depends on the model of UPS. An implementation of a [Management Daemon \(2.5\)](#) acting as a utility program may provide a listing of the variables available, as well as acting on them. Ed:
NUT: upscmd

Command	Meaning
beeper.disable	Disable the UPS beeper
beeper.enable	Enable the UPS beeper
beeper.mute	Temporarily mute the UPS beeper
load.off	Turn off the load immediately
load.off.delay	Turn off the load with a delay (seconds)
load.on	Turn on the load immediately
load.on.delay	Turn on the load with a delay (seconds)
shutdown.return	Turn off the load and return when power is back
shutdown.stayoff	Turn off the load and remain off
shutdown.stop	Stop a shutdown in progress

Table 7: Typical Instant Commands

Appendix B. Change Log

This section is to be removed before publishing as an RFC.

Ed: To be removed on publication.

B.1. Changes in version 01

1. There is exactly [one newline](#) ([4.1](#)) at the end of commands and responses.
2. Added descriptions to variables in [Annex](#) ([A](#)).
3. Added clause [Event](#) ([5.2](#)).

B.2. Changes in version 02

1. Extended acknowledgments.
2. Added reference to possible use of RFC1628 between driver and [Attachment Daemon](#) ([2.1](#)).
3. Clarified response to command LIST CLIENT.

Author's Address

Roger Price (editor)
Network UPS Tools Project
La Gaude
France

Phone: [+33 493 24 43 57](tel:+33493244357)
Email: ietf@rogerprice.org