LoRaWAN® Firmware Management Protocol Specification TS006-1.0.0-rc4

NOTICE OF USE AND DISCLOSURE

Elements of LoRa Alliance specifications may be subject to third-party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a member of the LoRa Alliance). The Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third-party intellectual property rights.

This document and the information contained herein are provided on an “AS IS” basis and THE ALLIANCE DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT. IN NO EVENT WILL THE ALLIANCE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The above notice and this paragraph must be included on all copies of this document.

LoRa Alliance®
5177 Brandin Court
Fremont, CA 94538
United States

Note: LoRa Alliance® and LoRaWAN® are trademarks of the LoRa Alliance, used by permission. All company, brand and product names may be trademarks that are the sole property of their respective owners.

LoRaWAN® Firmware Management Protocol Specification TS006-1.0.0-rc4

Authored by the FUOTA Working Group of the LoRa Alliance Technical Committee

Technical Committee Chair and Vice-Chair:
A. YEGIN (Actility), O. SELLER (Semtech)

Working Group Chair:
J. CATALANO (Kerlink)

Editor:
J. CATALANO (Kerlink)

Contributors:
J. CATALANO (Kerlink), N. SORNIN (Semtech), J. SWETINA (NEC)

Version: 1.0.0-rc4
Date: May 28, 2021
Status: RELEASE CANDIDATE

Contents

Tables

1 Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in IETF Best Current Practice 14 (BCP14 [RFC2119] [RFC8174]) when, and only when, they appear in all capitals, as shown here.

The tables in this document are normative. The figures and notes in this document are informative.

Document titles are written as *LoRaWAN Link Layer Specification* and sections within a document are written as "Frequency-Hopping Beacon Transmission".

Commands are written ***PackageVersionReq***, bits and bit fields are written `PackageIdentifier`, constants are written RECEIVE_DELAY1, variables are written *N*.

In this document:

- The octet order for all multi-octet fields SHALL be little endian.
- EUI are 8-octet fields and SHALL be transmitted as little endian.
- By default, `RFU` bits are Reserved for Future Use and SHALL be set to 0 by the transmitter of the packet and SHALL be silently ignored by the receiver.

2 Introduction

This document proposes a standard application layer firmware management protocol allowing to query / manage the following end-device's properties.

- What is the version of the firmware running on the end-device?
- What is the end-device's hardware version?
- Reboot the end-device at a given time
- Is an image available on the end-device for firmware upgrade, and what version?
- Delete an image

3 Firmware Management Package

The `PackageIdentifier` of the firmware management package is 4. The `PackageVersion` of this package is version 1.

The following messages are sent to each end-device individually, using unicast downlink on a port specifically used for the firmware management package. It is RECOMMENDED that a default port value of 203 be used to promote interoperability, though implementations MAY use a value from those not assigned by the LoRa Alliance®.

These messages SHALL NOT be sent using multicast. If these messages are received on a multicast address the end-device SHALL drop them silently.

These commands SHALL be invoked using one of the solutions described in *LoRaWAN Multi Package Access Protocol Specification* [TS007].

The following table summarizes the list of firmware management commands.

CID	Command name	Transmitted by		Short Description
		End-device	App Server	
0x00	***PackageVersionReq***		x	Used by the Application Server to request the package version implemented by the end-device
0x00	***PackageVersionAns***	x		Conveys the answer to ***PackageVersionReq***
0x01	***DevVersionReq***		x	Asks an end-device for its hardware version, and its currently running firmware version
0x01	***DevVersionAns***	x		Conveys the answer to the ***DevVersionReq*** request
0x02	***DevRebootTimeReq***		x	Instructs an end-device to program a reboot at a given time or immediately
0x02	***DevRebootTimeAns***	x		Conveys the answer to ***DevRebootTimeReq***
0x03	***DevRebootCountdownReq***		x	Instructs an end-device to program a reboot after a certain period of time has elapsed
0x03	***DevRebootCountdownAns***	x		Conveys the answer to ***DevRebootCountdownReq***
0x04	***DevUpgradeImageReq***		x	Asks an end-device if a firmware upgrade image is present in memory, its version and status
0x04	***DevUpgradeImageAns***	x		Conveys answer to the ***DevUpgradeReq*** request
0x05	***DevDeleteImageReq***		x	Instructs the end-device to delete a given firmware upgrade image
0x05	***DevDeleteImageAns***	x		Conveys answer to the ***DevDeleteImageReq*** request

Table 1: Firmware Management commands summary

3.1 Package Version Commands (*PackageVersionReq*, *PackageVersionAns*)

The ***PackageVersionReq*** command has no payload.

The end-device SHALL answer this command with a ***PackageVersionAns*** command with the following payload.

Size (octets)	1	1
Field	`PackageIdentifier`	`PackageVersion`

Table 2: PackageVersionAns

`PackageIdentifier` uniquely identifies the package.

`PackageVersion` corresponds to the major version of the package specification implemented by the end-device.

3.2 Device Version Commands (*DevVersionReq*, *DevVersionAns*)

The ***DevVersionReq*** command has no payload.

The end-device SHALL respond to this command with a ***DevVersionAns*** command with the following payload:

Size (octets)	4	4
Field	`FW version`	`HW version`

Table 3: DevVersionAns

The `FW  version` and `HW  version` fields are device manufacturer specific and freely allocated.

> **Example**: the `HW  version` field might be composed of a 24bits hardware identifier followed by an 8 bits revision index of this hardware platform.
>
> **Example**: the `FW version` field might be a 32bits CRC of the firmware currently running on the end-device.

3.3 Device Reboot Time Commands (*DevRebootTimeReq*, *DevRebootTimeAns*)

The ***DevRebootTimeReq*** command is used by the Application Server to instruct an end-device's application to program a hardware reboot either immediately or at a given time. If a valid firmware upgrade image is present at the time of reboot this upgrade SHALL be installed and the end-device SHALL reboot with a new firmware version.

The command's payload is:

Size (octets)	4
Field	`RebootTime`

Table 4: DevRebootTimeReq

RebootTime encodes the time at which the reboot SHALL take place.

A value > 0 but < 0xFFFFFFFF of RebootTime encodes the point in time, expressed in seconds since GPS epoch, *i.e.* Sunday January the 6th 1980 at midnight, (refer to the *LoRaWAN Link Layer Specification* [TS001] ***DeviceTimeReq*** MAC command) at which reboot takes place.

A value of 0x00000000 instructs the end-device to reboot as soon as possible.

A value of 0xFFFFFFFF instructs the end-device to cancel a currently programmed reboot.

The end-device SHALL only store a single programmed reboot event, meaning that any received valid ***DevRebootTimeReq*** command or ***DevRebootCountdownReq*** command overwrites all previously received reboot commands.

Behavior of the end-device upon reception of a ***DevRebootTimeReq*** command:

- If the RebootTime field has a value of 0x00000000 the end-device SHALL reboot as soon as possible. If the RebootTime field has a value > 0 but < 0xFFFFFFFF:
    1. If the end-device knows the current time and if
        - The value of the RebootTime field indicates a time in the future then the end-device SHALL reboot at that time.
        - The value of the RebootTime field indicates a time in the past then the end-device SHALL report an error with a ***DevRebootTimeAns*** command with a value of RebootTime of 0x00000000.
    2. If the end-device does not know the current time but is capable of measuring duration and is activated on a LoRaWAN 1.0.3 or later compatible Network Server then the end-device MAY request the current time from the Network Server – using the ***DeviceTimeReq*** command (refer to the *LoRaWAN Link Layer Specification* [TS001] ***DeviceTimeReq*** MAC command) – and proceed as in case (1).
    3. In all other cases the end-device SHALL report an error with a ***DevRebootTimeAns*** command with a value of RebootTime of 0x00000000.
- If the RebootTime field has a value of 0xFFFFFFFF the end-device SHALL cancel a currently programmed reboot and SHALL acknowledge with a ***DevRebootTimeAns*** command with a value of RebootTime of 0xFFFFFFFF.
- If a valid firmware upgrade image is present at the end-device, then that image SHALL be installed at reboot.

DevRebootTimeAns command:

If the end-device reboots following a 0x00000000 command, it SHALL NOT answer with a ***DevRebootTimeAns*** command.

In all other cases, the end-device SHALL sends a ***DevRebootTimeAns*** command back with the following payload:

Size (octets)	4
Field	RebootTime

Table 5: DevRebootTimeAns

The RebootTime field of the answer encodes the number of seconds between sending the ***DevRebootTimeAns*** message until reboot.

The ***DevRebootTimeAns*** command can also indicate an error or acknowledge a reboot cancellation:

A value of 0x00000000 of the `RebootTime` field in the ***DevRebootTimeAns*** message indicates the inability of the end-device to reboot at the requested time.
A value of 0xFFFFFFFF of the `RebootTime` field in the ***DevRebootTimeAns*** message acknowledges the cancellation of a currently programmed reboot.

3.4 Device Reboot Countdown Commands (*DevRebootCountdownReq*, *DevRebootCountdownAns*)

The ***DevRebootCountdownReq*** command is used by the Application Server to instruct an end-device's application to program a hardware reboot after a given time period. If a valid firmware upgrade image is present at the time of reboot this upgrade SHALL be installed and the end-device SHALL reboot with a new firmware version.

The command's payload is:

Size (octets)	3
Field	`Countdown`

Table 6: DevRebootCountdownReq

A value > 0 but < 0xFFFFFF of the 24-bit integer `Countdown` field encodes the number of seconds after which, after reception of the ***DevRebootCountdownReq*** command, the end-device SHALL reboot.

> **Note**: the maximum allowed value 0xFFFFFE corresponds to 16777214 seconds or 194 days, 4 hours, 20 minutes and 14 seconds.

A value of 0x000000 instructs the end-device to reboot as soon as possible.
A value of 0xFFFFFF instructs the end-device to cancel a currently programmed reboot.

The end-device SHALL only store a single programmed reboot event, meaning that any received valid ***DevRebootTimeReq*** or ***DevRebootCountdownReq*** command overwrites all previously received reboot commands.

Behavior of the end-device upon reception of a ***DevRebootCountdownReq*** command:

- If the `Countdown` field has a value of 0x000000 the end-device SHALL reboot as soon as possible.
- If the `Countdown` field has a value > 0 but < 0xFFFFFF:
    - If the end-device is capable of measuring time duration in seconds the end-device SHALL reboot when the duration indicated by the `Countdown` field has elapsed.
    - If the end-device is not capable of measuring time duration in seconds the end-device SHALL report an error with a ***DevRebootCountdownAns*** command with a value of `Countdown` of 0x000000.
- If the `Countdown` field has a value of 0xFFFFFF the end-device SHALL cancel a currently programmed reboot and SHALL acknowledge with a ***DevRebootCountdownAns*** command with a value of `Countdown` of 0xFFFFFF.
- If a valid firmware upgrade image is present at the end-device, then that image SHALL be installed at reboot.

DevRebootCountdownAns command:
If the end-device reboots following a 0x000000 command, it SHALL NOT answer with a ***DevRebootCountdownAns*** command.

In all other cases the end-device SHALL send a ***DevRebootCountdownAns*** command back with the following payload:

Size (octets)	3
Field	`Countdown`

Table 7: DevRebootTimeAns

The `Countdown` field of the answer encodes the number of seconds between sending the ***DevRebootCountdownAns*** message until reboot.

The ***DevRebootCountdownAns*** command can also indicate an error or acknowledge a reboot cancellation:

A value of 0x000000 of the `Countdown` field in the ***DevRebootCountdownAns*** message indicates the inability of the end-device to reboot at the requested time.

A value of 0xFFFFFF of the `Countdown` field in the ***DevRebootCountdownAns*** message acknowledges the cancellation of a currently programmed reboot.

3.5 Device Upgrade Image Commands (*DevUpgradeImageReq*, *DevUpgradeImageAns*)

The ***DevUpgradeImageReq*** command is used to ask an end-device if a firmware upgrade image is present in its memory, the validity of that firmware upgrade image and its version. The command has no payload.

The end-device SHALL answer with a ***DevUpgradeImageAns*** message with the following payload.

Size (octets)	1	4
Field	`Status`	`nextFirmwareVersion` (conditional)

Table 8: DevUpgradeImageAns

Where

Bits	7:2	1:0
`Status` **Fields**	`RFU`	`UpImageStatus`

Table 9: DevUpgradeImageAns Status fields

The `UpImageStatus` field encodes the status of the firmware upgrade image according to the following table.

value	`UpImageStatus`
0	No firmware upgrade image currently present
1	A firmware upgrade image is present but is either corrupted or its cryptographic signature is wrong
2	An authenticated firmware upgrade image is present but does not corresponds to the end-device's hardware platform

3	A valid firmware upgrade image is present and can be installed

Table 10 : UpImageStatus field encoding

The nextFirmwareVersion field SHALL be present only when UpImageStatus = 3, in that case it encodes the firmware version that will be running once this firmware upgrade image is installed.
Installation of the firmware upgrade image SHALL be performed when the end-device reboots. After successful reboot with a new firmware its firmware upgrade image SHALL be deleted from the end-device and SHALL no longer be reported.

> **Note 1**: Any (valid or invalid) firmware upgrade image (UpImageStatus 1, 2 or 3) is expected to be overwritten next time a firmware upgrade image download takes place.
>
> **Note 2**: Download and storage of the firmware upgrade image at the end-device is out of scope of the current specification.

3.6 Device Delete Image Commands (*DevDeleteImageReq*, *DevDeleteImageAns*)

The ***DevDeleteImageReq*** command is used to delete a firmware upgrade image currently stored in an end-device. The command payload is:

Size (octets)	4
Field	FirmwareToDeleteVersion

Table 11: DevDeleteImageReq

The end-device SHALL answer with a ***DevDeleteImageAns*** message with the following payload.

Size (octets)	1
Field	Status

Table 12: DevDeleteImageAns

Where

Bits	7:2	1	0
Status Fields	RFU	ErrorInvalidVersion	ErrorNoValidImage

Table 13: DevDeleteImageAns Status fields

The ErrorNoValidImage bit signals that the end-device has currently no valid firmware upgrade image stored.

The ErrorInvalidVersion bit signals that the version specified in the delete request FirmwareToDeleteVersion field does not match the version of the valid firmware upgrade image currently stored.
If any of those two bits is 1, the delete command did not succeed, the state of the end-device SHALL stay unmodified.
Upon reception of a ***DevDeleteImageReq*** command, a valid, uncorrupted, and authenticated firmware upgrade image that corresponds to the end-device's hardware platform and that has

a firmware version equal to `FirmwareToDeleteVersion` SHALL be deleted or invalidated by the end-device.

4 Glossary

CRC	Cyclic Redundancy Check
EUI	Extended Unique Identifier
FM	Firmware Management
FUOTA	Firmware Update Over-The-Air
FW	Firmware
GPS	Global Positioning System
HW	Hardware
RFU	Reserved for Future Use

5 Bibliography

5.1 References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[RFC8174] Leiba, B., “Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words”, BCP 14, RFC8174, May 1997

[TS001] LoRaWAN® 1.0.4 Link Layer Specification, LoRa Alliance, October 2020

[TS003] LoRaWAN® Application Layer Clock Synchronization Specification TS003-2.0.0, LoRa Alliance, TBD

[TS004] LoRaWAN® Fragmented Data Block Transport Specification TS004-2.0.0, TBD

[TS005] LoRaWAN® Remote Multicast Setup Specification TS005-2.0.0, LoRa Alliance, TBD

[TS007] LoRaWAN® Multi Package Access Protocol Specification TS007-1.0.0, LoRa Alliance, to be published