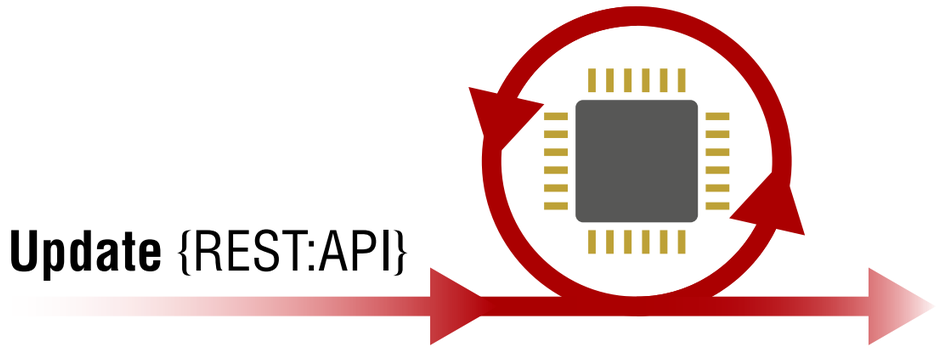
visual engineering
LIGHTWARE

# Application Notes

**Lightware Update REST API for UCX/MMX2 Devices**



# Table of Contents



## Document Information

**Document revision**: 1.1
**Release date**: 06-04-2023
**Editor:** Laszlo Zsedenyi

## Contact Us

sales@lightware.com
+36 1 255 3800

support@lightware.com
+36 1 255 3810

**Lightware Visual Engineering PLC.**
Peterdy 15, Budapest H-1071, Hungary
www.lightware.com

# 1

# Lightware Update REST API

**This document is about the updating process for UCX/MMX2 devices with Lightware Update REST API. This protocol allows integrating the update process into a third-party environment.**

## 1.1. Introduction

The **Update REST API (REpresentational State Transfer Application Public Interface)** is developed to have a standardized update interface between the Lightware device and a third-party software tool (e.g. external controller). The update means – generally – to install a new Lightware Firmware Package (LFP2) to the Lightware device. The Update REST API is part of the entire **Lightware REST API** software architecture.

The Update REST API is available over **HTTP/HTTPS and optionally** with basic authentication (with setting a password for the user admin).

## 1.2. Device Identification – Optional

The following commands help get basic information from the desired device.

### 1.2.1. Querying the Product Name of the Device

This is a **read-only** parameter, which cannot be changed.

**Request and Response**

➜ request: GET·http://<ip/host>/api/ProductName

🡐 response: <standard_response>
🡐 body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | <message>: <Product_name> The request has succeeded; the product name of the device is sent as text/plain content. |

**Example**

➜ request: GET http://192.168.0.114/api/ProductName

🡐 response: 200 OK
🡐 body: MMX2-4x3-H20

### 1.2.2. Querying the Serial Number of the Device

This is a **read-only** parameter, which cannot be changed.

**Request and Response**

➜ request: GET·http://<ip/host>/api/SerialNumber

🡐 response: <standard_response>
🡐 body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | <message>: <Serial_number> The request has succeeded; the unique S/N of the device is sent as text/plain content. |

**Example**

➜ request: GET http://myDevice/api/SerialNumber

🡐 response: 200 OK
🡐 body: D1000001

### 1.2.3. Querying the Device Label

**Request and Response**

➜ request: GET·http://<ip/host>/api/V1/MANAGEMENT/LABEL/DeviceLabel

🡐 response: <standard_response>
🡐 body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | <message>: <custom_name> The request has succeeded; the custom name of the device is sent as text/plain content. |

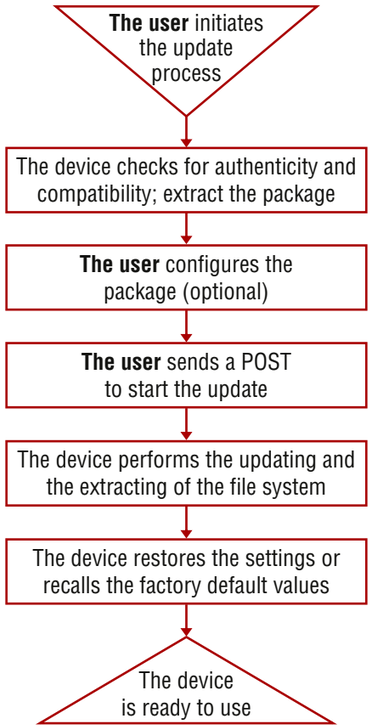
**Example**

➜ request: GET http://192.168.0.114/api/V1/MANAGEMENT/LABEL/DeviceLabel

🡐 response: 200 OK
🡐 body: MyDevice

## 1.3. The Updating Process

The API allows the device to update itself with an **LFP2** FW package. This new approach means there is no need to run a software on the connected PC (in case of LDU2) for updating the Lightware device.




***The Updating Process***

**The Updating Steps**

1. The **User uploads the package** to a dedicated URL.
   – The package is being unpacked during the uploading process.
   – The package is checked by the device to see if they are compatible with each other.
2. The **User can set the uploaded package**.
   – The factory default settings can be recalled (this is the default setting of the package) or the current settings of the device can be restored (optional).
3. The **User starts the self-updating process** by sending a POST method.
   – The device performs the operations in connection with the update (e.g. file system replacement).
   – The device restores the previous settings to the new system, if it was set to.
4. The device is ready to use.

### 1.3.1. Uploading the Package

This step may take up to one or two minutes to finish as the size of the LFP2 package is approx. 100-150 MB (depends on the device and FW version).

**Request and Response**

➜ request: POST·http://<ip/host>/api/V1/MANAGEMENT/UPDATE/Package
➜ body: <LFP2 file>
➜ content-type: application/octet-stream

➜ response: <standard_response>
➜ body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | <message>: **OK** The request has succeeded; the whole package is extracted and the basic checks are done. |
| | | **400 Bad request** | <message>: **Package incompatible: Incompatible partnumber: [...]** The request cannot be processed; error during the uploading/checking. e.g. the package is not compatible with the device. The <message> is in text/plain format. |
| | | **500 Internal Server Error** | <message>: **Socket timeout** The package uploading was not successful. Please try it again. |

**Example**

➜ request: POST http://192.168.0.114/api/V1/MANAGEMENT/UPDATE/Package
➜ body: (the LFP2 file)

➜ response: 200 OK
➜ body: OK

### 1.3.2. Configuration Parameters – Keep the Current Settings

**DIFFERENCE:** This feature is available from **FW package v2.1.0**.

**ATTENTION!** Restoring the device settings is not allowed if the firmware is **downgraded** to a previous version or the existing (=running) FW package of the device is **v1.x**. In those cases, recalling the factory default settings is a must.

This is an **optional step** that allows keeping the current settings of the device. The **default setting** of an update is to **recall the factory default** values. If you want to keep the current settings of the device, delete the **'dofactoryreset'** file from the uploaded package as follows:

**Request and Response**

➜ request: DELETE·http://<ip/host>/api/V1/MANAGEMENT/UPDATE/CONFIG/dofactoryreset

➜ response: <standard_response>
➜ body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | "Deleted" |
| | | **405 Method not allowed** | LFP2 package was not uploaded to the device. |
| | | **500 Internal Server Error** | The file does not exist in the uploaded package (e.g. it was deleted previously). Continue the update process. |

**Example**

➜ request: DELETE http://192.168.0.114/api/V1/MANAGEMENT/UPDATE/CONFIG/dofactoryreset

➜ response: 200 OK
➜ body: Deleted

### 1.3.3. Executing the Update

The command is for installing the uploaded package in place of the current system. This step may take up to 7 minutes.

**Request and Response**

➜ request: POST·http://<ip/host>/api/V1/MANAGEMENT/UPDATE/update
➜ body: <payload>
➜ content-type: text/plain

➜ response: <standard_response>
➜ body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <payload> | optional | **force** | The update can be forced to run in **Package incompatible** error state, but the security check cannot be bypassed (**Package untrusted** state). |
| <standard_response> | Standard HTTP response | **200 OK** | The request has succeeded; the update process finished successfully. |
| | | **400 Bad request** | The request cannot be processed. it may happen that the uploaded package is not compatible with the device (**Package incompatible** error state) and the **force** payload was **not** used. |

**ATTENTION!** Please note that running a **forced update** can make the device inoperable.

**Example**

➜ request: POST http://192.168.0.114/api/V1/MANAGEMENT/UPDATE/update

➜ response: 200 OK
➜ body: OK

When the *'force'* payload is used:

➜ request: POST http://192.168.0.114/api/V1/MANAGEMENT/UPDATE/update
➜ body: force
➜ content-type: text/plain

➜ response: 200 OK
➜ body: OK

### 1.3.4. Querying the State of the Updating Process

This command is for querying the current state of the updating process.

**Request and Response**

➜ request: GET·http://<ip/host>/api/V1/MANAGEMENT/UPDATE/Status

➜ response: <standard_response>
➜ body: <message>

**Parameters**

| Parameter | Parameter description | Values | Value description |
|---|---|---|---|
| <ip/host> | The IP address or the host name of the device. | | e.g.: 192.168.0.110, myDevice |
| <standard_response> | Standard HTTP response | **200 OK** | The request has succeeded.<br><message> states:<br>**Idle**: not active, update has not started yet.<br>**Package uploading**: updating and checking is in progress.<br>**Package ready**: uploading is done, checking is successful. |
| | | **400 Bad Request** | The request cannot be processed.<br><message> states:<br>**Package incompatible**: the uploaded package is not compatible with the device.<br>**Package untrusted**: the uploaded package is not secure.<br>**Failed**: the update is not successful. |

INFO: Please note that while the new firmware is being programmed in the device, it cannot send a response for a request. Thus the current state cannot be requested either.

After the firmware update is finished successfully, the device would send the 'Idle' response when requesting its status with this command.

**Example**

➜ request: GET http://192.168.0.114/api/V1/MANAGEMENT/UPDATE/Status

➜ response: 200 OK
➜ body: Package uploading