ads-tec IIT GmbH

# IRF1000 IRF2000 IRF3000

## Application Note – Firmware Update via API

| Version | Date | Editor | Changes |
|---|---|---|---|
| 1.0 | 12.04.2023 | SnPr | Creation |
| 1.1 | 22.11.2023 | SnPr | Small Fixes |

Table of Contents



# 1 Introduction

The ads-tec IIT industrial firewalls can be updated by different APIs and methods. This document gives a summary and examples.

There are three main methods:

1. All IRF1000, IRF3000 and IRF2000 (>= firmware version 2.5.0) can be updated using the JSON/RPC API and an additional FTP, TFTP or HTTP server from which the device will pull the firmware. The valid user accounts for using this method can be configured using the Permissions feature
2. All IRF1000, IRF3000 and IRF2000 (>= firmware version 2.6.7) can be updated using the JSON/RPC API and an HTTP server for automatic or manual background updating on slow links using small chunk files instead of the big monolithic firmware file. The user account for this method can be configured using the Permissions feature, but only via API. This method is used by the Big-LinX Device Management.
3. All devices can be updated by a classic and custom HTTP POST of the firmware file directly to the flashing application. This method is only allowed for the “admin” user account and only HTTP digest authentication is available. This method will be disabled by default in future firmware versions due to its historical authentication approach.

Of course the first two API controlled methods can be implemented as well using the ads-tec classic HTTP API (which is not recommended any more) or the ads-tec detection protocol ADSDP if a very low level protocol has to be used due to restricted development environments.

In general all updates are applied to a secondary partition. The bootloader will switch to the new firmware if the flashing was completed.
There are however some corner cases in which unique non-atomic sequences must be applied for very small time slots. Like seldom automatic bootloader updates. Thus a power loss while the update is applied should be avoided. A very small risk of an invalid configuration or firmware file on the device remains if a power loss occurs while the updated is in progress.

# 2 JSON/RPC API controlled update from server

## 2.1 JSON/RPC in general

Please take the ads-tec document “IRF3000 IRF2000 IRF1000 Application Note - ads-tec JSON RPC API specification v34-EN.pdf” as reference on how to transmit configuration variable changes and status queries.

## 2.2 Description

This method is historically integrated in all ads-tec products since our first router IF1000. The flasher application might shut down secondary services with high RAM usage depending on the device to free buffer RAM to load the file.
The device will reboot automatically as soon as the flashing is completed. There are no dedicated status calls for gathering the progress of the download or the flashing. Only the generic “status meminfo” can be used (see bellow)
There is no retransmission or retry in case of network interrupt or timeouts.

## 2.3 Config Variables

The following variables are in use.

| Name | Value | Description |
|---|---|---|
| update_proto | (http\|ftp\|tftp) | Selection of the protocol which shall be used to pull the update |
| update_server | IP or Hostname | IP address or hostname of the server which provides the file |
| update_restoredefaults | (enabled\|disabled) | Activates factory defaults on update. By default: disabled |
| fw_update_now | Any | Triggers the update |

# 3 JSON/RPC API controlled background update from HTTP server using the chunk files

## 3.1 JSON/RPC in general

Please take the ads-tec document “IRF3000 IRF2000 IRF1000 Application Note - ads-tec JSON RPC API specification v34-EN.pdf” as reference on how to transmit configuration variable changes and status queries.

## 3.2 Description

Starting with IRF1000, IRF3000 and IRF2000 2.6.5 an update of the second firmware partition in the background using small chunk files has been introduced.
With this mechanism, software components with high RAM consumption are not shut down!
A reboot into this new firmware does NOT take place, this must be done by the user or via the API through the reboot variables. The query which firmware version is ready on the second partition will be described in the following chapter.
For IRF2000 version < 4.0.0 the API behavior differs due to the deprecated OSGi features. The described behavior is for versions >= 4.0.0.

## 3.3 Config Variables

| Name | Wert | Beschreibung |
|---|---|---|
| fw_update_url | URL for the „Firmware Description File“ | Default value is: https://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/fldf-v2.txt, this value should always be transmitted as the ads-tec Big-LinX server might overwrite this if the Update |

| | | |
|---|---|---|
| | | by using the Big-LinX device management is in use as well. You can also replace this with your own server.<br>HTTPS can not be used, only the Big-LinX server cert will be accepted on HTTPS! The configured HTTP proxy will be used! |
| fw_update_url_blx | URL for the „Firmware Description File“ | Same as above but will be used if Big-LinX VPN is enabled and thus can be a service on the private Big-LinX VPN segment. Status Call blxstat_vpnstate must be “CONNECTED”.<br>HTTPS can not be used, only the Big-LinX server cert will be accepted on HTTPS.<br>The configured HTTP proxy will be used as well if the IP is not 10.237.0.1 (internal Big-LinX Firmware server IP on Big-LinX VPN) |
| fw_auto_download_now | any | Trigger the check and download of the chunk file into the second partition. |
| activate_update_now | any | Reboot to the new firmware on the second partition if available. |

## 3.4 Status Queries

| Name | Parameter 1 | Parameter 2 | Return Value | Description |
|---|---|---|---|---|
| firmwareupdate_prepared | - | | „yes“ or „no“ | “yes” if the firmware is valid and can be activated |
| firmwareupdate_progress | - | | -1 or 0-100 | Percent completed flashing or -1 if no flashing is active |
| firmwareupdate_inprogress | - | | “yes” or “no” | “yes” if an update flashing is in progress |
| diag_log | fw_update | | inernal error messege in case of failures | Can contain HTTP errors or any other internal error message which might happen during update. Can be used for debugging but should not be parsed programmatically as content, behaviour and syntax can change on newer versions. |

## 3.5 The fldf.txt file

The fldf.txt will be requested from the given server URL with the device AX number as a HTTP GET postfix “?sn=<AX……..>. Thus one can implement an active service which delivers different files for different devices.

Ads-tec hosts the fldf.txt file by default on the following location:
`https://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/fldf-v2.txt`

This is the default value of the variable fw_update_url.

For example, right now the file look like this:

```
# Firmware Location Description File Format 0.2
# 08.12.2014 Update IRF2xxx
Latest stable Firmware; IRF2xxx; 4.4.0; 150458; http://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/Ads-tec-IRF2xxx-4.4.0-SVN-R49835.B-150458.bin
Latest stable Firmware; IRF1xxx; 1.5.0; 150459; http://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/Ads-tec-IRF1xxx-1.5.0-SVN-R49835.B-150459.bin
Latest stable Firmware; IRF3xxx; 1.2.0; 150462; http://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/Ads-tec-IRF3xxx-1.2.0-SVN-R49835.B-150462.bin
```

## 3.6 The Chunk Files

The chunk files differ from the main firmware file which is distributed by ads-tec. If a custom server shall be used one must download them as a zip file and extract them on the own custom server URL.
The zip file is currently always downloadable under the same URL as given in our fldf.txt file but with the ending “zip” instead of bin. For example:

`https://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/Ads-tec-IRF3xxx-1.2.0-SVN-R49835.B-150462.zip`

The device itself will not pull the zip file but the single chunks which are hosted by ads-tec on a URL like this:

`https://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/Ads-tec-IRF3xxx-1.2.0-SVN-R49835.B-150462/000000.bin`

The same pattern has to be applied if the files shall be hosted on an own server.

# 4 Update by classic HTTP POST

## 4.1 Description

This method is historically implemented for all ads-tec devices since the IF1000. It is in use by the ads-tec web interface as well.
This method is only allowed for the “admin” user account and only HTTP digest authentication is available. This method will be disabled by default in future firmware versions due to its historical authentication approach. However it might be useful to upgrade old devices to the latest version.

## 4.2 HTTP POST to netflash

The main firmware file must be send by HTTP POST to the following device URL

```
http(s)://<IP>//cgi-bin/netflash
```

<IP> must be the address of the device, of cource a hostname is possible too. HTTPS should be used to encrypt the admin password on its way to the device.

The file must be encoded as HTTP multipart/form-data POST with a “file” form field.
As example this linux shell script snippet using curl and the variables $password, $ip and $filename will do the job:

```
curl -v --digest -u admin:$password https://$ip//cgi-bin/netflash?cgi://file, -F
file="@$filename"
```

# 5 Common Status Calls

The following status calls might be of interest in combination with firmware updates

| Name | Parameter 1 | Parameter 2 | Return Value | Description |
|---|---|---|---|---|
| firmware_version_downgrade_check | Version i.e. „1.5.0“ | - | „yes“ or „no“ | “yes” if the firmware downgrade is possible |
| product | - | - | i.e. „IRF2xxx“ for IRF2000 series devices | Firmware device family like it must be supplied in the fldt.txt file. |
| uptime | - | - | Unix “uptime” command | Useful for detecting reboots |
| configstate | - | - | i.e. “notsaved” | Useful for detecting unsaved changes |
| version | - | - | i.e. “1.5.0” | Returns the currently running main version string |
| buildno | - | - | String | Returns the detailed revision and build number |
| meminfo | netflash | - | Currently used resident RAM of the given process and the PID. i.e. “1234: 1000 kB” or empty if process does not exist. | Useful the check if the main flasher application is still running and downloading or flashing the file. (not useful for chunk updates!) |