OTA Upgrade Cluster

- The existing OTA upgrade doesn't supports the interoperabilty,i.e.,only manufacturer specific upgrade is available.
- The OTA upgrade cluster we are using is provides the features of interoperability using this any manufacturer upgrade image can be used by the any devices.

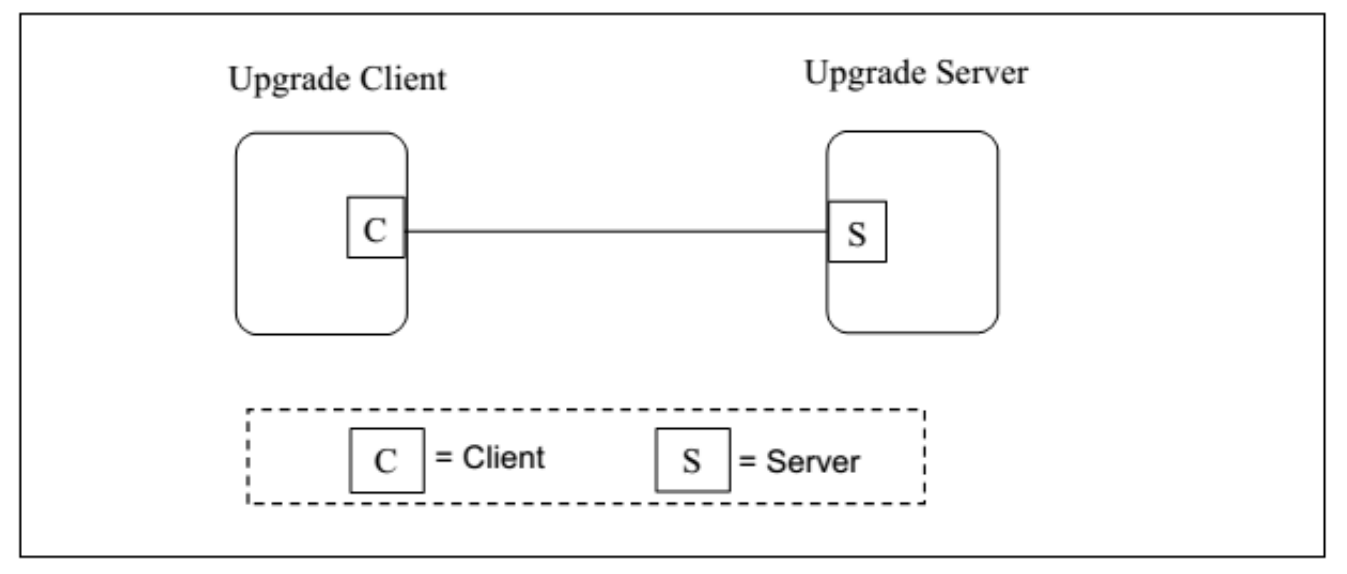
Cluster Name	Cluster ID	Description
OTA Upgrade	0x0019	Parameters and commands for upgrading image on devices Over The Air.

Definitions

- Application Bootloading:- Bootloading is the method where the device has memory for receiving the upgrade image while running current stack.
- OTA Upgrade Server:- The ZigBee device class that sends the commands and image to a client device as part of OTA upgrade process.
- OTA Upgrade Client:- The ZigBee device that is the target for receiving and upgrading its (running) image.
- Manufacturer:- A company or a group of companies that produce the device including both the software and the hardware

Typical Usage of OTA Cluster Upgrade

- As shown in figure the Upgrade Server Notify the Upgrade Client or Client send request for new image and the Upgrade process take place over the air.



- Using this OTA service not only new images but device specific files like security credential, log or configuration of device also send.

Overview

- Image Upgradation is done using OTA messages between two devices of different manufacturer . For this device required application bootloader and additional memory for receiving images.
- Here Upgrade server is responsible for notify the devices for arrival of new images, and also decide client's activity like, upgrade,downgrade or reinstall.
- ZR(ZigBee Router) devices will notify the image notify from server and the upgrade process continues.
- Whereas, ZEDs are sleepy device so there is case when server sends the image notify commads the ZEDs are is in sleepmode.
- For this ZED periodically polling for the new image is available o not.

Security

- Security for the OTA Upgrade cluster encompasses these areas:
  - image verification,
  - image transport and
  - image encryption.
- The OTA Upgrade cluster is intended to be used with all ZigBee application profiles. Security mechanism of each application profile dictates the security level of over-the-air image upgrading.
- For example application profile with strict security policies (such as Smart Energy) may support image signature as well as encryption in both network and APS layers; while Home Automation application profile may only support network encryption .

Image Verification

- Authenticity and Integrity of the bootload image is must be done. This is most often accomplished through asymmetric encryption technologies where only one device is able to create a digital signature but many devices are able to verify it.
- Bootload images may be signed by the private key of the manufacturer with that signature appended to the image that is transported to the device. Once the complete image has been received the signature is verified using the public key of the signer.
- Devices may have pre-installed with the certificate of the device that created the signature or may be received over the air.

Image Transport

- Transport of the images in secure fashion provides little additional security to the integrity and authenticity.
- What secure transportation provides is a *means to communicate the policies* about when a device should perform upgrade or what version it should upgrade to.
- A secured ZigBee network uses network security for all messages, but that does not provide point-to-point security. APS security should be used to assure that messages are sent from only the trusted source (the upgrade server)
- Point-to-point security is not provided by broadcast and multicast,so the image receiving and upgrading processes are split up in two events. Upgrading is done using unicast in which sever is authorized the client to upgrade the received image with former one or not.

Image Signature

- For certain application profiles, the OTA Upgrade cluster ***provides mechanisms to sign the OTA file*** to protect the authenticity and integrity of the image.
- The application profiles, must determine if this is required.

OTA File Format

- The OTA file format is composed of a header followed by a number of sub-elements.
- The header describes general information about the file such as version, the manufacturer that created it, and the device it is intended for.
- Sub-elements in the file may contain upgrade data for the embedded device, certificates, configuration data, log messages, or other manufacturer specific pieces.

Octets	Variable	Variable	Variable	Variable
Data	OTA Header	Upgrade Image	Signer Certificate	Signature

OTA Header

- As shown in table many fields of the OTA Header itself give its description.
- Some of the field needs explanation which is given in following slides.
- The endianness used in each data field shall be little endian in order to be compliant with general ZigBee messages.

Octets	Data Types	Field Names	Mandatory/Optional
4	Unsigned 32-bit integer	OTA upgrade file identifier	M
2	Unsigned 16-bit integer	OTA Header version	M
2	Unsigned 16-bit integer	OTA Header length	M
2	Unsigned 16-bit integer	OTA Header Field control	M
2	Unsigned 16-bit integer	Manufacturer code	M
2	Unsigned 16-bit integer	Image type	M
4	Unsigned 32-bit integer	File version	M
2	Unsigned 16-bit integer	ZigBee Stack version	M
32	Character string	OTA Header string	M
4	Unsigned 32-bit integer	Total Image size (including header)	M
0/1	Unsigned 8-bit integer	Security credential version	O
0/8	IEEE Address	Upgrade file destination	O
0/2	Unsigned 16-bit integer	Minimum hardware version	O
0/2	Unsigned 16-bit integer	Maximum hardware version	O

OTA Header Fields

- **OTA Header Field Control**:- The bit mask indicates whether additional information such as Image Signature or Signing Certificate are included as part of the OTA Upgrade Image.

Bits	Name
0	Security Credential Version Present
1	Device Specific File
2	Hardware Versions Present
3 - 15	Reserved

OTA Header Fields

- **Manufacturer Code**:-This is the ZigBee assigned identifier for each member company. When used during the OTA upgrade process, manufacturer code value of **0xffff** has a special meaning of a wild card **"match all"** effect.
- **Image Type:-** The manufacturer should assign an appropriate and unique image type value to each of its devices order to distinguish the products.

This is a manufacturer specific value. However, the OTA Upgrade cluster has reserved the last 64 values of image type value to indicate specific file types.

File Type Values	File Type Description
0x0000 – 0xffbf	Manufacturer Specific
0xffc0	Security credential
0xffc1	Configuration
0xffc2	Log
0xffc3 – 0xfffe	Reserved (unassigned)
0xffff	Reserved: wild card

OTA Header Fields

- **File Version**:- For firmware image, the file version represents the release and build number of the image's application and stack. The application release and build numbers are manufacturer specific.

Application Release	Application Build	Stack Release	Stack Build
1 byte	1 byte	1 byte	1 byte
8-bit integer	8-bit integer	8-bit integer	8-bit integer

- **File version A**: 0x10053519 represents application release 1.0 build 05 with stack release 3.5 b19.
- **File version B:** 0x10103519 represents application release 1.0 build 10 with stack release 3.5 b19.
- **File version C:** 0x10103701 represents application release 1.0 build 10 with stack release 3.7 b01.
- C >> B >>A

OTA Header Fields

- **ZigBee Stack Version**:-This information indicates the ZigBee stack version that is used by the application.

ZigBee Stack Version Values	Stack Name
0x0000	ZigBee 2006
0x0001	ZigBee 2007
0x0002	ZigBee Pro
0x0003	ZigBee IP
0x0004 – 0xffff	Reserved

OTA Header Fields

- **Security Credential Version:-**This information indicates security credential version type, such as SE1.0 or SE2.0 that the client is required to have, before it shall install the image.

Security Credential Version Values	Security Credential Version Types
0x00	SE 1.0
0x01	SE 1.1
0x02	SE 2.0
0x03 – 0xff	Reserved

OTA Header Fields

- **Total Image Size:-** The value represents the total image size in bytes shall be transferred over-the-air from the server to the client.

In most cases, the total image size of an OTA upgrade image file is the sum of the OTA header and the actual file data (along with its tag) lengths. If the image is a signed image and contains a certificate of the signer, then the Total image size shall also include the signer certificate and the signature (along with their tags) in bytes.

- **Upgrade File Destination:-** If *Device Specific File bit* is set, it indicates that this OTA file contains security credential/certificate data or other type of information that is specific to a particular device.

Hence, the upgrade file destination field (in OTA header) should also be set to indicate the IEEE address of the client device that this file is meant for.

OTA Header Fields

- **Minimum Hardware Version**:-The value represents the earliest hardware platform version this image should be used on.
- **Maximum Hardware Version:-**The value represents the latest hardware platform this image should be used on. The field is defined the same as the Minimum Hardware Version (above).

Version	Revision
1 byte	1 byte
8-bit integer	8-bit integer

- The hardware version of the device should not be earlier than the minimum (hardware) version and should not be later than the maximum (hardware) version in order to run the OTA upgrade file

Sub-element Format

- Sub-elements in the file are composed of an identifier followed by a length field, followed by the data.
- The identifier provides for forward and backward compatibility as new sub-elements are introduced. Existing devices that do not understand newer sub-elements may ignore the data.

Octets	2-bytes	4-bytes	Variable
Data	Tag ID	Length Field	Data

Tag Identifiers	Description
0x0000	Upgrade Image
0x0001	ECDSA Signature
0x0002	ECDSA Signing Certificate
0x0003 – 0xefff	Reserved
0xf000 – 0xffff	Manufacturer Specific Use

ECDSA Signature & Certificatae Sub-element

- The ECDSA Signature sub-element contains a signature for the entire file as means of insuring that the data was not modified at any point during its transmission from the signing device.

Octets	2-bytes	4-bytes	8-bytes	42-bytes
Data	Tag ID: 0x0001	Length Field: 0x00000032	Signer IEEE Address	Signature Data

- This shall contain the data for the ECDSA certificate of the device

Octets	2-bytes	4-bytes	48-bytes
Data	Tag ID: 0x0002	Length Field: 0x00000030	ECDSA Certificate

Attributes

- Attributes defined for OTA Upgrade cluster is shown in table.
- Currently, all attributes are client side attributes (**only stored on the client)**.
- There is **no server side** attribute at the moment.
- All attributes with the exception of **UpgradeServerID** should be initialized to their default values before being used

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*UpgradeServerID*	IEEE Address	-	Read	0xffffffffffffffff	M
0x0001	*FileOffset*	Unsigned 32-bit integer	0x00000000 – 0xffffffff	Read	0xffffffff	O
0x0002	*CurrentFileVersion*	Unsigned 32-bit integer	0x00000000 – 0xffffffff	Read	0xffffffff	O
0x0003	*CurrentZigBeeStackVersion*	Unsigned 16-bit integer	0x0000 – 0xffff	Read	0xffff	O
0x0004	*DownloadedFileVersion*	Unsigned 32-bit integer	0x00000000 – 0xffffffff	Read	0xffffffff	O
0x0005	*DownloadedZigBeeStackVersion*	Unsigned 16-bit integer	0x0000 – 0xffff	Read	0xffff	O
0x0006	*ImageUpgradeStatus*	8-bit enumeration	0x00 – 0xff	Read	0xff	O
0x0007	*Manufacturer ID*	Unsigned 16-bit integer	0x0000-0xffff	Read	-	O
0x0008	*Image Type ID*	Unsigned 16-bit integer	0x0000-0xffff	Read	-	O

ImageUpgradeStatus Attribute

- The upgrade status of the client device. The status indicates where the client device is at in terms of the download and upgrade process.
- The status helps to indicate whether the client has completed the download process and whether it is ready to upgrade to the new image.

Image Upgrade Status Values	Description
0x00	Normal
0x01	Download in progress
0x02	Download complete
0x03	Waiting to upgrade
0x04	Count down
0x05	Wait for more
0x06 – 0xff	Reserved

OTA Cluster Parameters

- Below are defined parameters for OTA Upgrade cluster server. These values are considered as parameters and not attributes because their values tend to change often and are not static.

Name	Type	Range	Default	Mandatory / Optional
QueryJitter	Unsigned 8-bit integer	0x01 – 0x64	0x32	M
DataSize	Unsigned 8-bit integer	0x00 – 0xff	0xff	M
OTAImageData	Octet	Varied	all 0xff's	M
CurrentTime	Unsigned 32-bit integer	0x00000000 – 0xffffffff	0xffffffff	M
UpgradeTime or RequestTime	Unsigned 32-bit integer	0x00000000 – 0xffffffff	0xffffffff	M

Query Jitter Parameter

- The parameter is part of Image Notify Command sent by the upgrade server. The parameter indicates whether the client receiving Image Notify Command should send in Query Next Image Request command or not.
- The server chooses the parameter value between 1 and 100 (inclusively) and includes it in the Image Notify Command. On receipt of the command, the client will examine other information to determine if they match its own values.
- If the information matches then, Client by randomly choosing a number between 1 and 100 and comparing it to the value of the QueryJitter parameter received.
- If it is less than or equal to the QueryJitter value from the server, it shall continue with the query process. If not, then it shall discard the command and no further processing shall continue.

CurrentTime and UpgradeTime/RequestTime Parameters

- If **CurrentTime and UpgradeTime** are used in the command (ex. Upgrade End Response), the server uses the parameters to notify the client when to upgrade to the new image.
- If **CurrentTime and RequestTime** are used in the command (ex. Image Block Response), the server is notifying the client when to request for more upgrade data.
- The CurrentTime indicates the current time of the OTA server.
- The UpgradeTime indicates the time that the client shall upgrade to running new image.
- The RequestTime indicates when the client shall request for more data.

OTA Upgrade Diagram

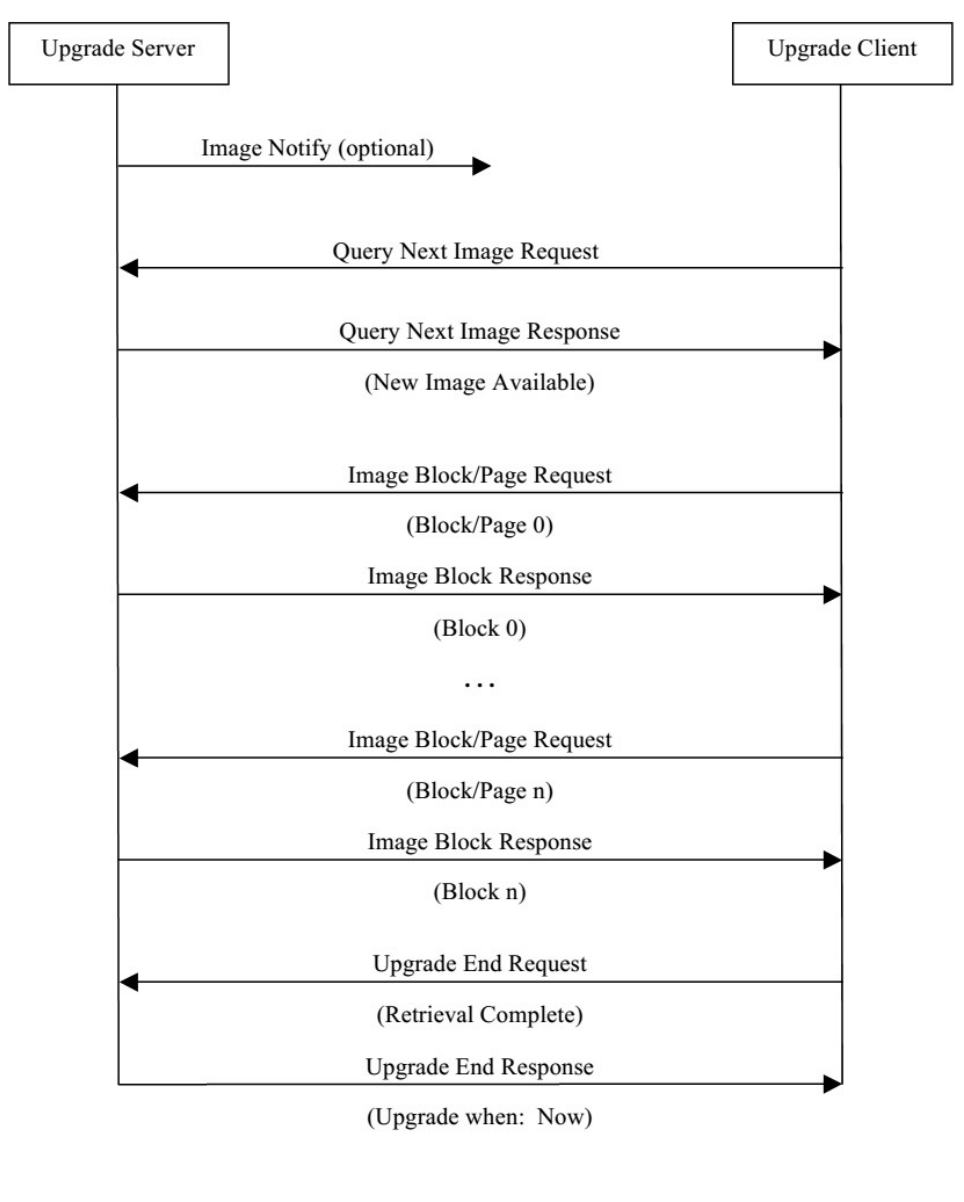


Image Notify Command

- The purpose of sending Image Notify command is so the server has a way to notify client devices of when the OTA upgrade images are available for them.
- It eliminates the need for ZR client devices having to check with the server periodically of when the new images are available.
- However, all client devices still need to send in Query Next Image Request command in order to officially start the OTA upgrade process.

Octets	**1**	**1**	**0/2**	**0/2**	**0/4**
Data Type	8-bit Enumeration	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit
Field Name	Payload type	Query jitter	Manufacturer code	Image type	(new) File version

Image Notify Command

Payload Type Values	Description
0x00	Query jitter
0x01	Query jitter and manufacturer code
0x02	Query jitter, manufacturer code, and image type
0x03	Query jitter, manufacturer code, image type, and new file version
0x04 – 0xff	Reserved

- **When Generated**:- Image Notify is sent from sever to client in unicast,multicast or broadcast manner for new upgrade image. Here not using unicast the it's purely not authoritative. On the payload type value it will send appropriate parameters.
- **Effect of Receipt**:-On receipt of a unicast Image Notify command, the device shall always send a Query Next Image request back to the upgrade server. This provides a way for the server to force reinstallation of image on the device.

On receipt of a broadcast or multicast Image Notify command, the device shall keep examining each field included in the payload with its own value.

Query Next Image Request Command

- Payload Format:-

Octets	1	2	2	4	0/2
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 16-bit
Field Name	Field control	Manufacturer code	Image type	(Current) File version	Hardware version

- The field control indicates whether additional information such as device's current running hardware version is included as part of the Query Next Image Request command.
- The 0th bit decides Hardware Version is present or not in Frame.

Query Next Image Request Command

- **When Generated:-** Client devices shall send a Query Next Image Request command to the server to see if there is new OTA upgrade image available. ZR devices may send the command after receiving Image Notify command. ZED device shall periodically wake up and send the command to the upgrade server. Client devices query what the next image is, based on their own information.
- **Effect on Receipt:-**The server takes the client's information in the command and determines whether it has a suitable image for the particular client. The server send back a response that indicates the availability of an image that matches the client info. And choose to upgrade, downgrade, or reinstall clients' image, as its policy dictates.

Query Next Image Response Command

- Payload Format:-

Octets	1	0/2	0/2	0/4	0/4
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 32-bit
Field Name	Status	Manufacturer code	Image type	File version	Image size

- Only if the status is SUCCESS that other fields are included. For other (error) status values, only status field shall be present

Query Next Image Response Command

- **When Generated:-**The upgrade server sends a Query Next Image Response with one of the following status: SUCCESS,NO_IMAGE_AVAILABLE or NOT_AUTHORIZED. When a SUCCESS status is sent, it is considered to be the explicit authorization to a device by the upgrade server that the device may upgrade to a specific software image.
- **Effect on Receipt:-**A status of SUCCESS in the Query Next Image response indicates to the client that the server has a new OTA upgrade image. The client shall begin requesting blocks of the image using the Image Block Request command. A ZED client may choose to change its wake cycle to retrieve the image more quickly.

Image Block Request Command

- **Payload Format :-**

Octets	1	2	2	4	4	1	0/8
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 32-bit	Unsigned 8-bit	IEEE Address
Field Name	Field control	Manufacturer code	Image type	File version	File offset	Maximum data size	Request node address

- Field control value is used to indicate additional optional fields that may be included in the payload of Image Block Request command. Currently, the device is only required to support field control value of 0x00; support for other field control value is optional.
- 0^{th} bit of the Field Control Device the Optional IEEE Address is present or not.

Image Block Request Command

- **When Generated:-**The client device requests the image data at its leisure by sending Image Block Request command to the upgrade server. The client knows the total number of request commands it needs to send from the image size value received in Query Next Image Response command.
- **Effect on Receipt:-**The server uses the manufacturer code, image type, and file version to uniquely identify the OTA upgrade image request by the client. It uses the file offset to determine the location of the requested data within the OTA upgrade image.

Image Page Request Command

- **Payload Format :-**

Octets	1	2	2	4	4	1	2	2	0/8
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 32-bit	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	IEEE Address
Field Name	Field control	Manufacturer code	Image type	File version	File offset	Maximum data size	Page size	Response Spacing	Request node address

- Field Control field is same as in previous command.
- Page size value indicates the number of bytes to be sent by the server before the client sends another Image Page Request command. In general, page size value shall be larger than the maximum data size value.
- Response Spacing Value indicates how fast the sever shall send data, upto this value server will wait before sending more data. Its in miliseconds.

Image Page Request Command

- **When Generated:-**The support for the command is optional. The client device may choose to request OTA upgrade data in one page size at a time from upgrade server. Using Image Page Request reduces the numbers of requests sent from the client to the upgrade server, compared to using Image Block Request command.

  In order to conserve battery life a device may use the Image Page Request command**.**

- **Effect on Receipt:-**The server uses the file offset value to determine the location of the requested data within the OTA upgrade image. The server may respond to a single Image Page Request command with possibly multiple Image Block Response commands; depending on the value of page size.

Image Block Response Command

- **Payload Format :-**

Image Block Response Command Payload with SUCCESS status

Octets	1	2	2	4	4	1	Variable
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 32-bit	Unsigned 8-bit	Octet
Field Name	Success status	Manufacturer code	Image type	File version	File offset	Data size	Image data

Image Block Response Command Payload with WAIT_FOR_DATA status

Octets	1	4	4
Data Type	Unsigned 8-bit	Unsigned 32-bit	Unsigned 32-bit
Field Name	Wait for data Status	Current time	Request time

Image Block Response Command

Image Block Response Command Payload with ABORT status

Octets	1
Data Type	Unsigned 8-bit
Field Name	Abort Status

- Image Block Response Status in the Image Block Response command may be SUCCESS, ABORT or WAIT_FOR_DATA. If the status is ABORT then only the status field shall be included in the message, all other fields shall be omitted.
- **When Generated:-**Upon receipt of an Image Block Request command the server shall generate an Image Block Response. If the server is able to retrieve the data for the client, it will respond with a status of SUCCESS and it will include all the fields in the payload.

  If status is differ then the payload also differ as per the status.

Image Block Response Command

- **Effect on Receipt :-**When the client receives the Image Block Response it shall examine the status field. If the value is SUCCESS it shall write the image data to its additional memory space.

Upgrade End Request Command

- **Payload Format :-**

Octets	1	2	2	4
Data Type	Unsigned 8-bit	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit
Field Name	Status	Manufacturer code	Image type	File version

- The status value of the Upgrade End Request command shall be SUCCESS, INVALID_IMAGE, REQUIRE_MORE_IMAGE, or ABORT.

Upgrade End Request Command

- **When Generated:-**Upon reception all the image data, the client should verify the image to ensure its integrity and validity.If the device requires signed images it shall examine the image and verify the signature. And if the checks fails then send INVALID_IMAGE status, otherwise send SUCCESS or REQUIRE_MORE_IMAGE.
- **Effect on Receipt :-**Upgrade End Request command does not have disable default response bit set. Hence, in a case where the Upgrade End Request command has been received and the server does not send Upgrade End Response command in response, a default response command shall be sent with SUCCESS status.

Upgrade End Response Command

- **Payload Format :-**

Octets	2	2	4	4	4
Data Type	Unsigned 16-bit	Unsigned 16-bit	Unsigned 32-bit	Unsigned 32-bit	Unsigned 32-bit
Field Name	Manufacturer code	Image type	File version	Current time	Upgrade time

- The ability to send the command with **wild card** values for manufacturer code, image type and file version is useful in this case because it eliminates the need for the server having to send the command multiple times for each manufacturer.

Upgrade End Response Command

- **When Generated :-**When an upgrade server receives an Upgrade End Request command with a status of INVALID_IMAGE, REQUIRE_MORE_IMAGE, or ABORT, no additional processing shall be done in its part.

  If the upgrade server receives an Upgrade End Request command with a status of SUCCESS,then it will send response with required fields and also tell client when to Upgrade New Image.

- **Effect on Receipt:-**The client shall examine the manufacturer code, image type and file version to verify that they match its own.

OTA Upgrade Process

- Once a device has completely downloaded the image and returned a status of SUCCESS in the Upgrade End Request, it shall obey the server's directive based on when it should upgrade.
- If the response directs client to wait forever then the client shall periodically query about Upgrade Time in every 60 minutes and after three successive failure it may apply image by itself.
- If the connection is loss then client will try to rediscover and connect to sever and if not possible to connect and get response then it may apply the image.

OTA Upgrade Recovery

- Each manufacturer is encouraged to implement a recovery method that should be used to recover the node in a case when the OTA upgrade fails.
- The recovery method is particularly important in a case where the device may not be able to communicate to the server over-the-air.
- One option for recovery method is the ability for the application bootloader to swap the images between its external flash and its internal flash, rather than just overwriting the internal with the external.
- In a case where the device is no longer able to communicate to the server over-the-air; the application bootloader could revert to the previous image via a button press on power up.