BLE

Device Firmware Update over BLE

Example proprietary profile used to update firmware (application image) file using *Bluetooth* Low Energy. This is not a standard profile, and is defined by Nordic to demonstrate how a typical Device Firmware Update can be achieved.

Device Firmware Update Profile

Profile Dependencies

The Device Firmware Update (DFU) Profile is used to transfer an application image from a BLE central (example, PC or Smart Phone) to a peripheral (example, Heart Rate Sensor) that supports Device Firmware Updates using the DFU Service.

Profile Changes

The following is an overview of the main changes to the profile in order to support update of SoftDevice and Bootloader over BLE.

- A Request Op Code has been added when writing 'Start DFU' to the DFU Control Point.
  The Request Op Code specifies the update procedure method, i.e. SoftDevice, Bootloader, or Application, see **DFU Control Point** for details.
- When writing 'Image Size' to the DFU Packet Characteristic it is now mandatory to write 12 bytes (3 times uint32) to indicate size of images being transfered, see **DFU Packet** for details. Earlier versions of the DFU Service used 4 bytes (uint32)

Profile Dependencies

This profile requires the Generic Attribute Profile (GATT).

Configuration

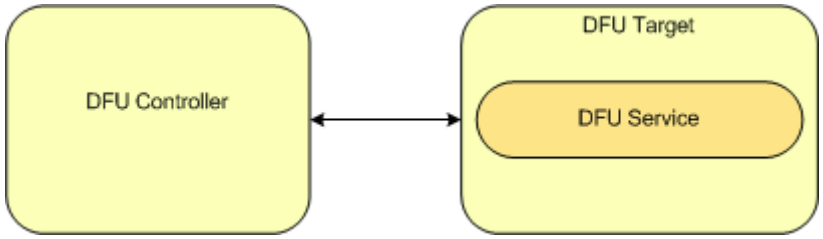
Roles

The profile defines two roles: DFU Target and Client. The DFU Target is the device which receives an image from the DFU Controller.

- The DFU Target, implements a GATT Server. This is the device under update.
- The DFU Controller, implements a GATT client. This will be the device that uploads a new firmware image to the DFU Target.

Role/Service Relationships

The following diagram shows the relationships between services and the two profile roles.



Relationship between profile roles and services

A DFU Target instantiates one instance of the DFU service.

DFU Controller Role Requirements

The DFU Controller shall support the **Device Firmware Update BLE Service**.

Service	DFU Controller
Device Firmware Update Service	M

General Error Handling.

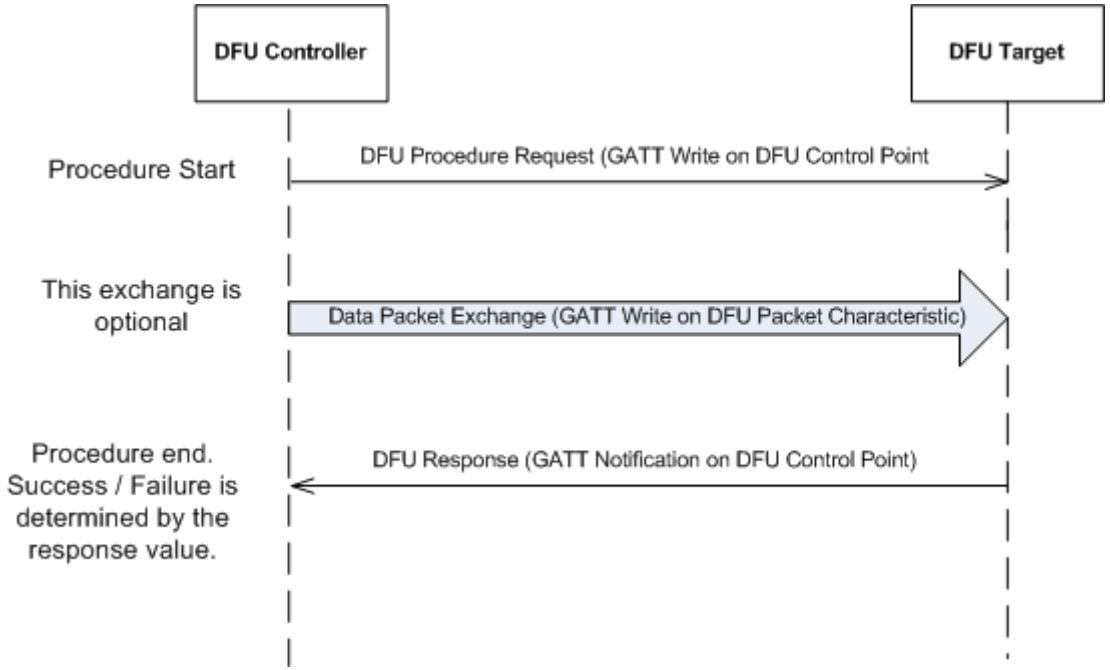
The DFU Controller shall also be tolerant when receiving the following ATT Error Code defined in CSS Part B, Section 1.2 of Supplement to the *Bluetooth* Core Specification, Version 3 or later:

- Client Characteristic Configuration Descriptor Improperly Configured

DFU Image Transfer Procedure

The DFU Controller requests one of the procedures defined for Device Firmware Update using the *DFU Control Point*. Each procedure requested by the Controller is characterised by a request on the DFU Control Point, using the GATT Write Procedure. This may be followed by additional writes on the DFU packet characteristic based, followed by end of procedure marked by a response from the DFU Target for the procedure using GATT Notifications on the Control Point. Response code in the response packet determines the success or failure of the procedure. In case of failure, the response codes provide an indication on possible reason for failure. Refer to the detailed description for more details in the *DFU Control Point*. The DFU Target process one request at a time, therefore if a request is received on the Target when already processing one, an ATT Error Response with 'Procedure Already In Progress' is received on the DFU Controller. Asynchronous Status and Image Size received information are sent by the DFU Target as GATT Notifications on the control point.

Below is small MSC to show a typical DFU Control Point Procedure



Typical DFU Control Point procedure

DFU Procedures defined are listed below:

DFU Procedure	Description
Start DFU	Procedure to prepare the device for a uploading the firmware. Size of firmware is provided in the request parameter. DFU Target response determines the device is ready for next stage of firmware update. The Start DFU must be followed by a byte indicating type of update,
Receive Init Data	procedure to exchange information necessary for firmware update that needs to be exchanged before upload of firmware. Device Type, Revision Type, Application version, Hash exchange / Public key exchange etc are some examples of such information. For detailed description of the init packet, see **Safety-checking the image**.
Receive App Data	Procedure to upload firmware fragmented as DFU Packets. Maximum size of each packet is (ATT_MTU - 3) octets. DFU Packets can be of variable length with a minimum size of 1 octet.
Validate	Procedure to validate updated firmware image. Current implementation supports only CRC, provided in "Receive Init Data" procedure, optional validation (size validation is done always)
Activate Image & Reset	Procedure to activate the new firmware and restart the device with new firmware. This procedure results in a GAP Disconnect.

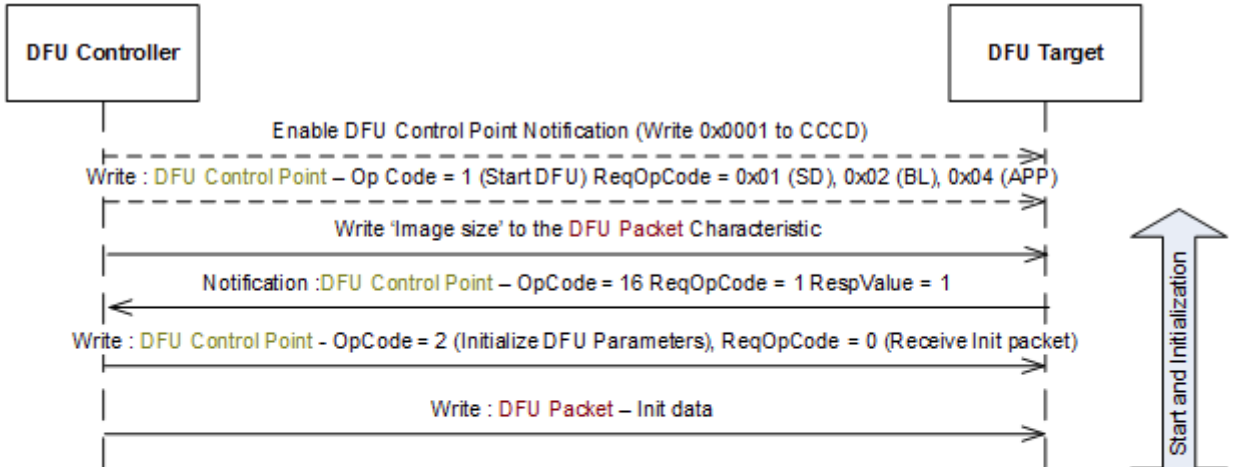
System Reset	Procedure to reset system. This procedure result in a GAP Disconnection Procedure.
Report Received Image Size	Procedure to request the size of image received.
Packet Receipt Notification	Procedure to request notification on receiving every 'n' packets. The number 'n' is requested by the controller. This is not a mandatory procedure for the Controller.

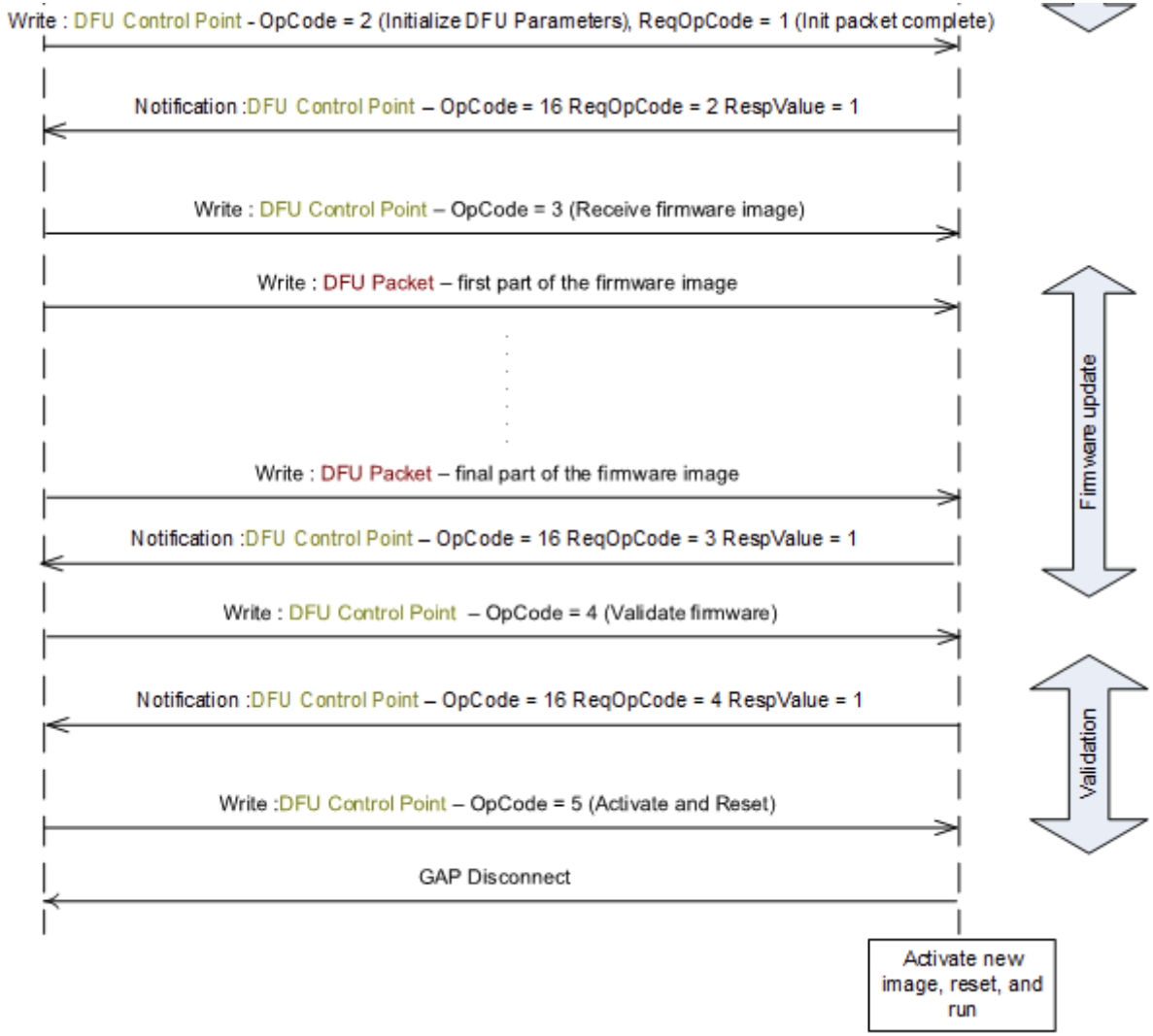
DFU Target in DFU mode, is in **GAP General Discoverable mode** and is **connectable**. 128-bit DFU Service UUID is advertised in the Advertisement Data, along with full name of **DfuTarg**.

Once connected to the DFU Controller, DFU Procedure can be started by requesting a Start DFU Procedure. Size of new firmware is provided during this procedure. On successful *Start DFU* procedure, *Receive Init Data* is initiated to exchange Init packet data information, **Safety-checking the image**. On success, this procedure is followed by upload of new firmware by requesting the *Receive App Data* Procedure. Currently, banked update is performed, implying, the old firmware can be restored in case firmware update procedure did not succeed at any stage of DFU Transfer. New firmware transferred is validated using the *Validate Procedure*. Validated firmware can be activated by issuing the *Activate Image & Reset Procedure*. It is possible to reset the system at any stage using the *System Reset* procedure. When this procedure is requested, the system will restart with old firmware in case the device had a firmware; in case the device did not have any existing application firmware the system will restart in DFU mode.

Firmware being updated is expected in binary format. It is possible to request the size of received firmware by requesting the *Report Received Image Size* procedure. This procedure is particularly useful on reconnection after a link loss. See **Link Loss Procedure** for more details. It is also possible for the Controller to request notification of acknowledge for every 'n' packets received using the *Packet Receipt Notification* procedure.

MSC below describes a typical firmware update exchange between a DFU Controller and DFU Target.





Transfer of an image to the DFU Target.

Idle Mode Procedures

In case no connection is established with the DFU Target after 60 seconds, Target device will restart in normal mode with old firmware, in case there was an existing application, else in DFU mode, in case there was no application on the device.

Image Validation Procedure

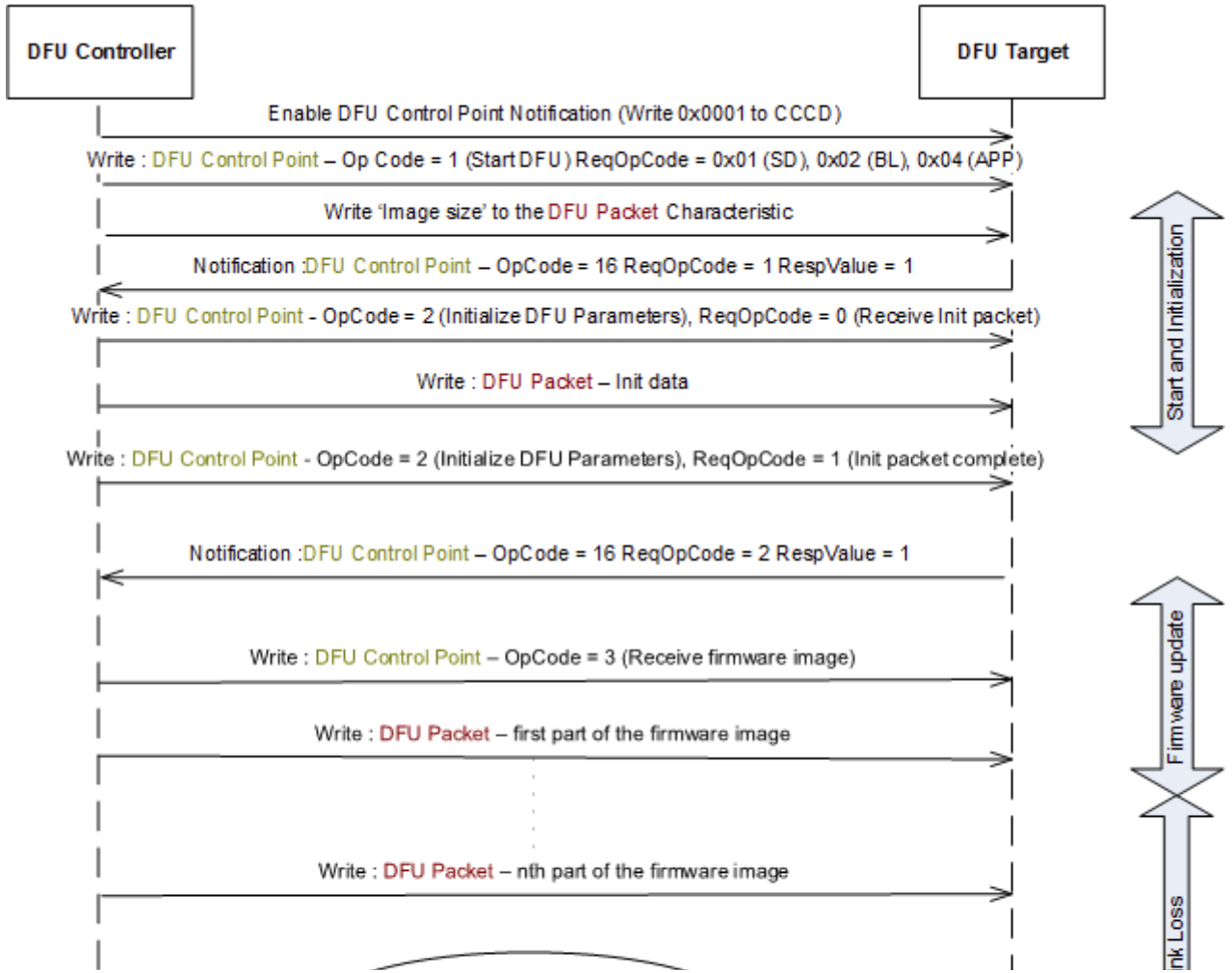
Procedure validates the updated firmware image. This is performed on the post validate function, **dfu_init_postvalidate**, in `dfu_init_template.c`.

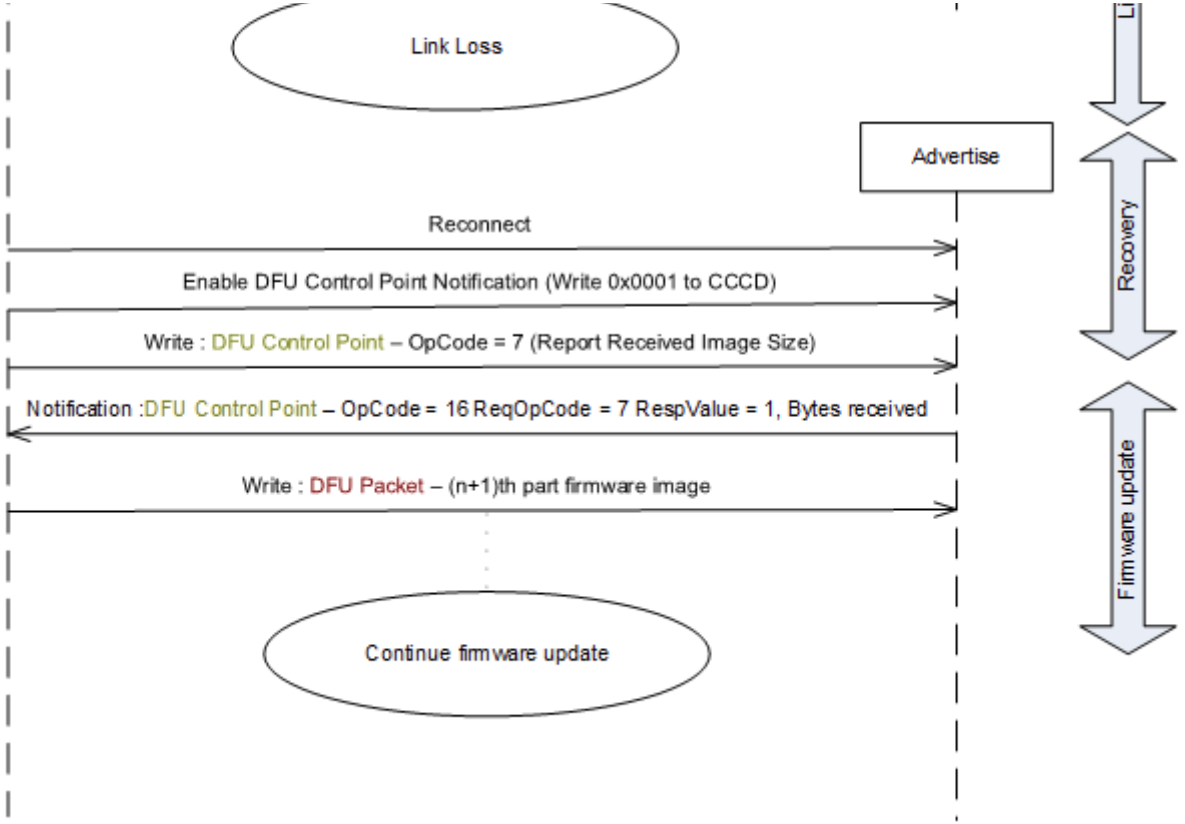
Idle Connection Procedure

When the Nordic bootloader application detects that the connection has been idle for a time period, meaning that the DFU Controller has not written any packet that will result in the progress of the firmware update, the update will be stopped, the connection will be terminated, and the previously valid application will be started. If no valid application exists in the flash, the application stays in bootloader mode waiting for a reconnection from the DFU Controller.

Link Loss Procedure

When the Nordic bootloader application detects a link loss, DFU state and image size is remembered. On reconnection to the Controller, the transfer can resume from where it had stopped. Controller can request size of received image using the *Report Received Image Size* Procedure and start transfer by applying the necessary offset. **Idle Mode Procedures** apply in the disconnected state on link loss.



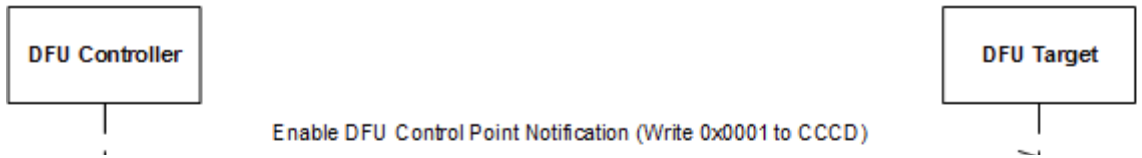


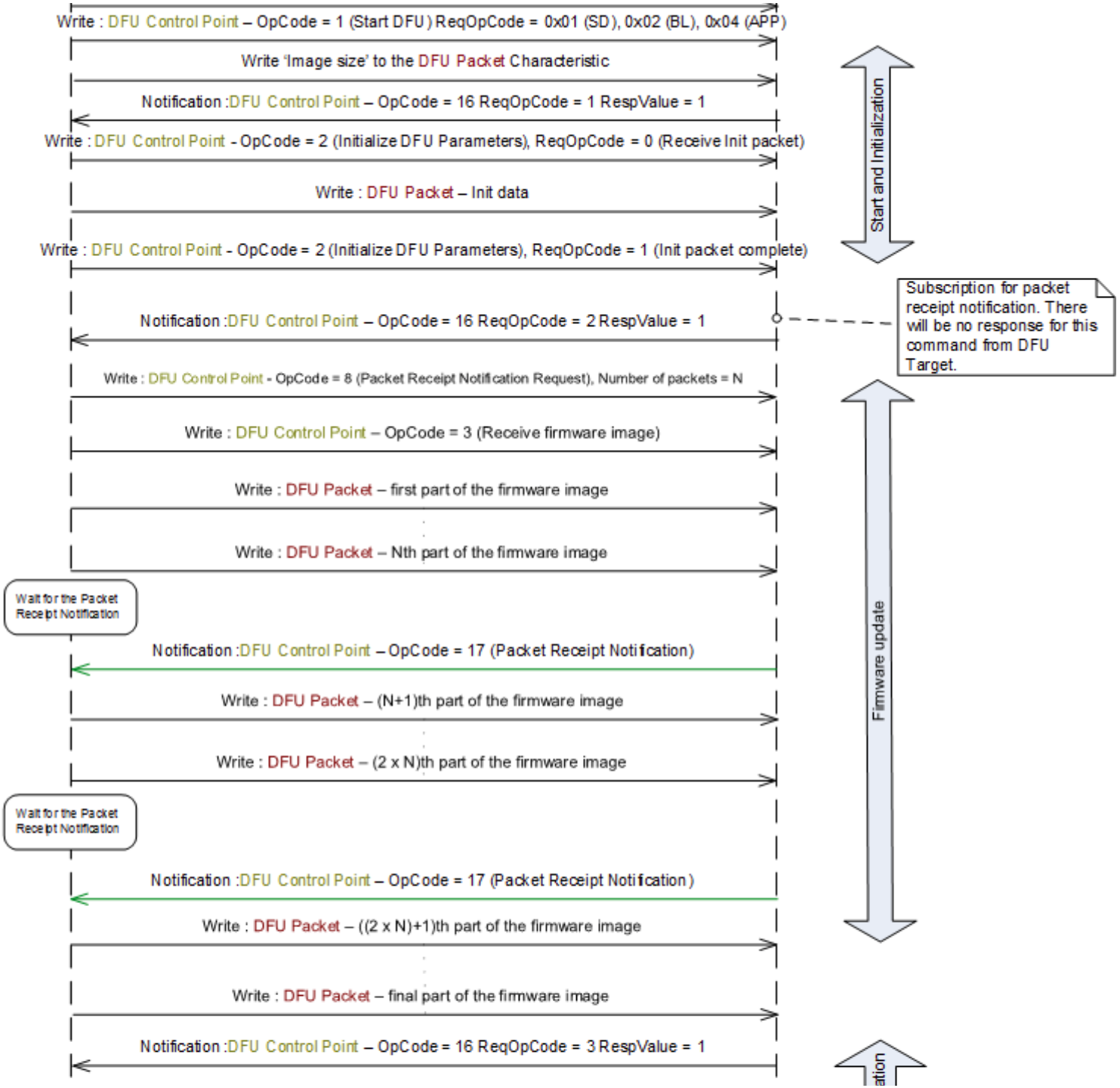
Recovery after Link Loss.

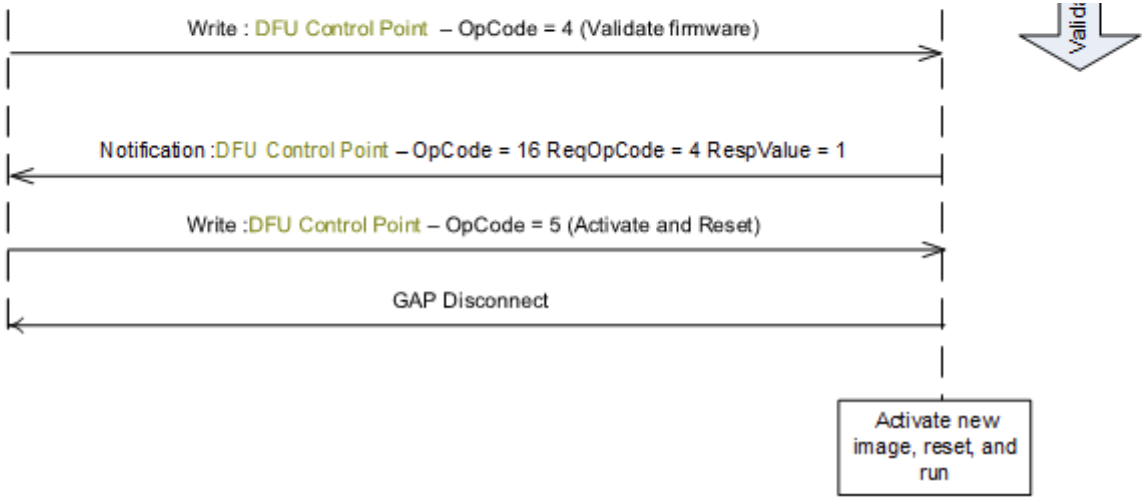
Packet Receipt Notification procedure

This feature allows the DFU Controller to subscribe for a notification from the DFU Target, each time a given number of firmware packets is received by the latter. To use this feature, the DFU Controller should write to the DFU Control Point, the Op Code 0x08 - *Packet Receipt Notification Request* followed by the *Number of packets*. If the *Number of packets* field is set to *N*, then the DFU Target will send a Notification of the DFU Control Point with Op Code = 0x11 - *Packet Receipt Notification*, each time it receives *N* number of firmware packets. The DFU Controller should wait for this notification each time it has finished sending *N* number of firmware packets.

In *Packet Receipt Notification*, the DFU Target also sends the number of bytes of firmware data received until that point of time. This may be used for any consistency checks by the DFU Controller.




Write : DFU Control Point – OpCode = 1 (Start DFU) ReqOpCode = 0x01 (SD), 0x02 (BL), 0x04 (APP)
Write 'Image size' to the DFU Packet Characteristic
Notification :DFU Control Point – OpCode = 16 ReqOpCode = 1 RespValue = 1
Write : DFU Control Point - OpCode = 2 (Initialize DFU Parameters), ReqOpCode = 0 (Receive Init packet)
Write : DFU Packet – Init data
Write : DFU Control Point - OpCode = 2 (Initialize DFU Parameters), ReqOpCode = 1 (Init packet complete)
Start and Initialization
Notification :DFU Control Point – OpCode = 16 ReqOpCode = 2 RespValue = 1
Subscription for packet receipt notification. There will be no response for this command from DFU Target.
Write : DFU Control Point - OpCode = 8 (Packet Receipt Notification Request), Number of packets = N
Write : DFU Control Point – OpCode = 3 (Receive firmware image)
Write : DFU Packet – first part of the firmware image
Write : DFU Packet – Nth part of the firmware image
Wait for the Packet Receipt Notification
Notification :DFU Control Point – OpCode = 17 (Packet Receipt Notification)
Write : DFU Packet – (N+1)th part of the firmware image
Write : DFU Packet – (2 x N)th part of the firmware image
Wait for the Packet Receipt Notification
Notification :DFU Control Point – OpCode = 17 (Packet Receipt Notification)
Write : DFU Packet – ((2 x N)+1)th part of the firmware image
Firmware update
Write : DFU Packet – final part of the firmware image
Notification :DFU Control Point – OpCode = 16 ReqOpCode = 3 RespValue = 1



Packet Receipt Notification procedure.

Security Considerations

This section describes the security considerations for a DFU Target and DFU Controller.

DFU Target Security Considerations

All supported characteristics specified by the DFU Service are set to Security Mode 1 and Security Level 1.

Device Firmware Update BLE Service

Overview

Nordic Device Firmware Update (DFU) Service exposes necessary information to perform Device Firmware Update on the device. NOTE: This is not a service defined by the *Bluetooth* SIG, a proprietary one to demonstrate a typical firmware update on an nRF51 device.

DFU Service does not depend on any other service. Support for following GATT sub-procedures are mandatory for this service: a. Write Characteristic Value b. Notifications c. Read Characteristic Descriptors d. Write Characteristic Descriptors

DFU GATT Service can operate on *Bluetooth* Low Energy as transport only.

The Service does not define any new error codes for Attribute Protocol and data exchange in is little endian (LSB first) order.

This service is instantiated as a primary service in the DFU mode.

Proprietary Service UUID

The Service UUID assigned is value 0x1531 over proprietary base, see table below.

UUID For Nordic:

Description	Number Base
Company Identifier:	0x0059
UUID Base:	0x23, 0xD1, 0xBC, 0xEA, 0x5F, 0x78, 0x23, 0x15, 0xDE, 0xEF, 0x12, 0x12, 0x00, 0x00, 0x00, 0x00
Service UUID start:	0x1530
Characteristic UUID start:	0x1531

Service Characteristics

DFU Service exposes one instance of characteristics listed in table below. This service does not impose any security requirements.

Characteristic Name	Requirement	Mandatory Properties	Descriptors	Description
DFU Packet	M	WriteWithoutResponse		See **DFU Packet**
DFU Control Point	M	Write, Notify		See **DFU Control Point**

DFU Packet

UUID: 0x1532 over proprietary base.

This characteristic receives firmware to nRF51 device as DFU Packets. The firmware is transferred by writing each fragment as DFU Packet to this characteristic. Size of each packet is in range of 1 to (ATT_MTU - 3). Packets must be in little endian (LSB first) order.

Names	Field Requirement	Format	Minimum Value	Maximum Value	Additional Information
DFU Packet	Mandatory	uint8	N/A	N/A	This field may be repeated up to a maximum of 20 times. In other words, the maximum length of this characteristic shall be 20 bytes.

DFU Packet

Note

When writing the 'Image Size' data to the Packet Characteristic after writing 'Start DFU' to the DFU Control Point this field must be written as:
<Length of SoftDevice><Length of Bootloader><Length of Application>
All lengths must be uint32. If a length is not present, e.g. when transferring only SoftDevice the value of the characteristic should be written as:
<Length of SoftDevice> 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

DFU Control Point

All DFU Procedures are requested using this characteristic. A DFU Procedure request is initiated by writing to this characteristic. A Response, marking end of procedure is received as a notification. The Characteristic value or the DFU Control PDU has the following format

UUID: 0x1531 over proprietary base.

The DFU Control Point characteristic is used to control the state of Device Firmware Update process.

Table below summarizes control point procedure op-code and respective parameters.

Names	Field Requirement	Format	Minimum Value	Maximum Value	Additional Information			
					Enumerations			
					Key	Value	Requirement	Description
					0	Reserved for future use		
					1	Start DFU	C4	Initiate the firmware update procedure. The response to this control point is a notification of the control point with Op Code 0x10, followed by the request Op Code 0x01, and the appropriate Response Value.
					2	Initialize DFU Parameters	C5	Prepare to receive init packets. The response to this control point is a notification of the control point with Op Code 0x10, followed by the request Op Code 0x02, and the appropriate Response Value.
					3	Receive firmware image		Prepare to receive the firmware data. The response to this control point is a notification of the control point with Op Code 0x10, followed by the request Op Code 0x03, and the

									request Op code 0x03, and the appropriate Response Value.
						4	Validate firmware		Validate the firmware received. Response to this control point is a notification of the control point with Op Code 0x10 followed by the appropriate Response Value.
	Op Code	Mandatory	uint8	N/A	N/A	5	Activate image and reset		Activate the previously received image and perform system reset. There is no response to this control point.
						6	Reset System		Perform system reset. There is no response to this control point.
						7	Report received Image Size		Request the DFU Target to report the total number of bytes of firmware data (excluding the start data and init data) received. The response to this control point is a notification of the Control Point with Op Code 0x10 followed by the request Op Code 0x07, the appropriate Response Value, and the number of bytes in the Response Parameter.
						8	Packet receipt notification request	C1	Request the DFU Target to enable/disable notification of Control Point characteristic each time the specified number of packets containing firmware data have been received. There is no response to this control point.
						16	Response Code	C2	The Response Code is followed by the Request Op Code, the Response Value and optionally, the Response Parameter.
						17	Packet Receipt Notification	C3	A notification sent by the DFU Target indicating that a new set of the preconfigured number of firmware data packets has been received.
						9-15	Reserved for future use		
						18-255	Reserved for future use		
	Number of								

packets **Information:** Parameter Value for "Enable periodic notification" Op Code	C1	uint16	N/A	N/A	Number of packets of firmware data to be received by the DFU Target before sending a new Packets receipt notification (Control Point notification with Op Code = 7). If this value is 0, then the notification of packets receipt will be disabled by the DFU Target.
Request Op Code **Information:** Parameter Value for "Response Code" Op Code	C2	uint8	N/A	N/A	Refer to the Op Code table above for additional information on the possible values for this field.
Response Value **Information:** C2: This Field is Mandatory for "Response Code" Op Code, otherwise this field is Excluded.	C2	uint8	N/A	N/A	Enumerations

Key	Value	Description
0	Reserved for future use	
1	Success	Response for successful operation
2	Invalid state	The DFU controller has performed an operation that is not valid in the current state of the firmware update process.
3	Not supported	The previous operation performed by the DFU Controller or the data sent by the DFU controller is not supported.
4	Data size exceeds limit	The DFU controller is trying to send more firmware data than expected.
5	CRC Error	A CRC error has occurred. Only used by Validate firmware procedure.
6	Operation failed	Response if the requested procedure failed.
7-255	Reserved for future use	

Response Parameter **Information:** C2: This Field is may be optionally present for	C2	variable	N/A	N/A	Note: The Response Parameter Value of the response to the Control Point is a variable length field to allow a list of different values defined by

present for "Response Code" Op Code, otherwise this field is Excluded.					the Service Specification
Number of bytes of firmware image received **Information:** C3. Present if the Op Code is 0x11 (Packet Receipt Notification).	C3	uint32	N/A	N/A	Number of bytes of firmware data (excluding the start and init data) received by the DFU Target at the given point of time.
DFU Image type **Information:** C4. Present if the Op Code is 0x01 (Start DFU). This field is parsed as bit field where each bit set indicates the image transferred for the requested DFU.	C4	uint8	N/A	N/A	Enumerations

Key	Value	Description
0x00	No image	No image will be updated
0x01	SoftDevice	A SoftDevice image will be transferred
0x02	Bootloader	A Bootloader image will be transferred
0x03	SoftDevice Bootloader	A SoftDevice w/ Bootloader image will be transferred
0x04	Application	An application image will be transferred
0x05-0x07	Other image combinations	Currently not supported
0x08-0xFF	Reserved for future use	

DFU Init packet **Information:** C5. Present if the Op Code is 0x02 (Init DFU packet).	C5	uint8	N/A	N/A	Enumerations

Key	Value	Description
0x00	Init packet Receive	Receive DFU init packet
0x01	Init packet complete	Transmission of DFU Init packet complete

DFU Control Point

General Error Handling procedures

If an Op Code is written to the *DFU Control Point* characteristic and the Client Characteristic Configuration descriptors of either or both of the *DFU Control Point* or the *DFU Status Report* are not configured for notifications, the DFU Target will return an error response with the Attribute Protocol Application error code set to *Client Characteristic Configuration Descriptor Improperly Configured* as defined in CSS Part B, Section 1.2 of Supplement to the *Bluetooth* Core Specification, Version 3 or later.