
# Kirale Binary Interface

## *Reference Guide*

### Introduction

This document describes the Kirale Binary Interface (KBI), an ultra-efficient binary communication protocol used by hosts to configure and interact with KiNOS through a UART port.

## Disclaimer

INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND DOES NOT REPRESENT A COMMITMENT ON THE PART OF KIRALE TECHNOLOGIES.

KIRALE TECHNOLOGIES PROVIDES THIS DOCUMENT “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF FITNESS OR MERCHANTABILITY FOR A PARTICULAR PURPOSE. KIRALE MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THIS MANUAL OR IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS MANUAL AT ANY TIME.

## Trademarks and Copyright



## Contact Information

Kirale Technologies S.L.
General Vara de Rey 9, 5B
26001 – Logroño (SPAIN)

+34 941 578 578
info@kirale.com
https://www.kirale.com

# Contents

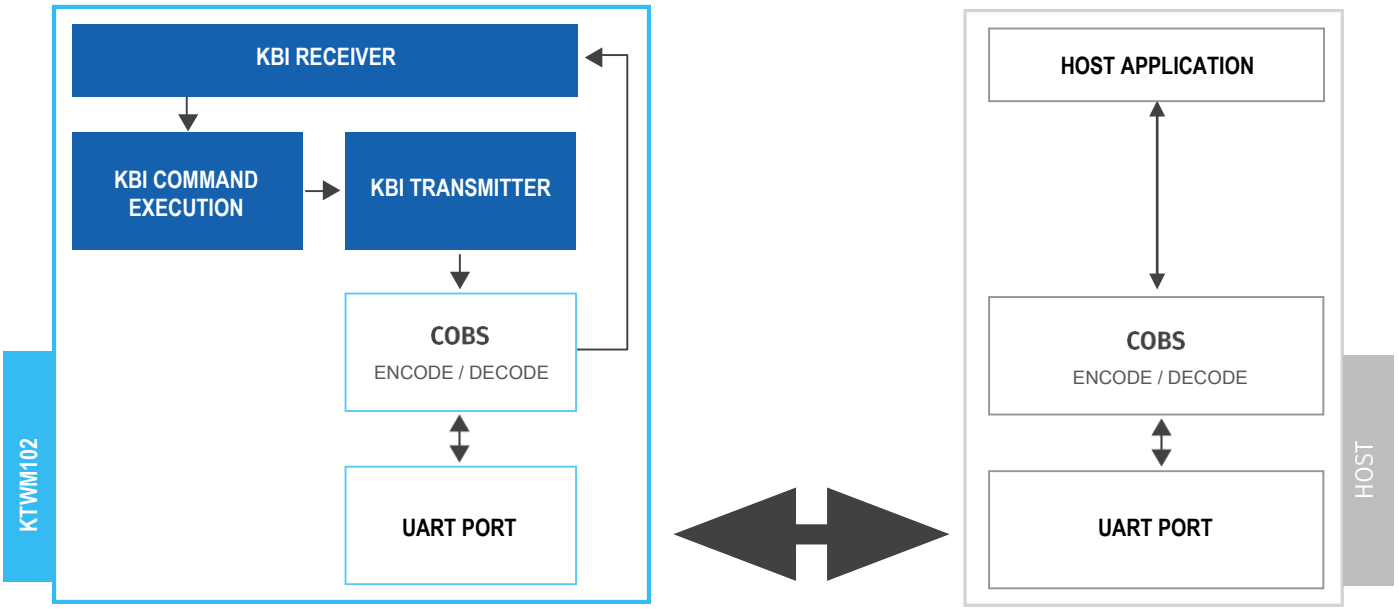
# Revision History

| Date | Revision | Changes |
|---|---|---|
| 03/2018 | 1.0 | Initial release. |
| 06/2019 | 2.0 | Mayor changes in KBI frame format. New commands and descriptions have also been added. |

# 1. Introduction to KBI Protocol

## 1.1. Interface operation

The picture below represents an external host using KBI to communicate with KiNOS via UART port:




ⓘ Serial configuration: 115200 bps, 8 data bits, no parity, 1 stop bit; no flow control.

⚠ Check that serial communication pins voltage complies with electrical characteristics.

As detailed in the picture, there is an intermediate coding/decoding level between the KBI and the UART interface. This is necessary to transmit a frame with the start delimiter character 0x00. Each time a delimiter character is transmitted/received it indicates the beginning of a new frame. For that reason other characters 0x00 contained in the frame need to be encoded to not confuse them as a new start delimiter. If an error is detected in UART frame reception and receiver is unable to decode the message, an error notification (encoded as [0x00 0xFF]) is automatically transmitted to sender.

This procedure is done by the Consistent Overhead Byte Stuffing (COBS) coding system implemented in KiNOS and which is explained in next section. So an external host application needs to implement this feature in order to be able to interact with KiNOS via UART.

## 1.2. Byte Stuffing

The purpose of byte stuffing is to convert data packets into a form suitable for transmission over a serial medium. When packets are sent over a serial medium there needs to be some way to tell where one packet ends and the next begins, particularly after errors, and this is typically done by using a special reserved value to indicate packet boundaries.

Byte stuffing ensures, at the cost of a potential increase in packet size, that this reserved value does not inadvertently appear in the body of any transmitted packet. In general, some overhead (additional bytes transmitted over the serial medium) is inevitable if we are to perform byte stuffing without loss of information.

A little effort to minimize the worst-case overhead has been made in this algorithm called **Consistent Overhead Byte Stuffing** (**COBS**). Refer to draft-ietf-pppext-cobs-00 for more detailed information.

### *1.2.1 Encode/decode basis*

As mentioned before, in KiNOS the special reserved value is **0x00** byte and it is used as UART frame start delimiter. COBS algorithm first takes the data and logically appends a single zero byte at the end (it is not necessary that the encoding routine actually adds this zero byte to the end of the packet in the memory, it simply has to behave as if the added zero was there).

Then COBS locates all the zero bytes in the packet (including the added one), and divides the packet at these boundaries into zero-terminated chunks. Every zero-terminated chunk contains one zero byte and it is always at the end of the chunk (trailing zero). A chunk may be as short as one byte (i.e. a chunk containing just a solitary zero byte) or as long as an entire frame.

Finally, COBS encodes (replaces) each zero-terminated chunk using one **COBS code block** followed by the chunk **non-zero bytes** and eliminates the zero byte. This method makes that all zero bytes contained in the frame disappear and then only a zero is added at the beginning as (special reserved value) start delimiter of the frame.  This encoding is totally

reversible using a COBS decode function.

Apart from that, KiNOS also implements Zero-Pair and Zero-Run Elimination (COBS extensions) in addition of the basic COBS to improve byte compression when two or more zeros are found together in the frame. So the COBS (ZPE-ZRE) code blocks used by KiNOS are:

| Code (n) | Followed by | Meaning |
|---|---|---|
| 00 | | Unused (Start delimiter) |
| 01 - CF | n - 1 data bytes | n - 1 bytes plus one implicit trailing zero. |
| D0 | n - 1 data bytes | n - 1 bytes with no implicit trailing zero. *(207 data bytes with no zero at the end)* |
| D1 - D2 | | Unused. |
| D3 - DF | nothing | A run of (n-D0) zeros (max. 15 zeros). |
| E0 - FE | n – E0 data bytes | n - E0 bytes plus two trailing zeros. *(*max. *30 data bytes)* |
| FF | | Used to error signal. |

### *1.2.2. Example*

Ping command (raw data):

```
00  12  40  06  44  FD  A5  2A  9E  CC  18  00  00  00  00  00  FF  FE  00  00  01  00  28
```

**Header** **Payload**

- Step 1 (add implicit zero at the end):

```
00  12  40  06  44 FD  A5  2A  9E  CC  18  00  00  00  00  00  FF  FE  00  00  01  00  28  00
```

  - Added for codification (not in real frame)

- Step 2 (divide in chunks with one or more zeros at the end):

```
00 | 12  40  06  44  FD  A5  2A  9E  CC  18  00  00 | 00  00  00 | FF  FE  00  00 | 01  00 | 28  00
```

- Step 3 (chunk 1 encoded):

```
01                                                    ; No data bytes plus one trailing zero at the end.
```

- Step 4 (chunk 2 encoded):

```
EA  12  40  06  44  FD  A5  2A  9E  CC  18            ; 10 data bytes plus two trailing zeros at the end.
```

- Step 5 (chunk 3 encoded):

```
D3                                  ; A run of 3 zero bytes.
```

- Step 6 (chunk 4 encoded):

```
E2  FF  FE                          ; Two data bytes plus two trailing zeros at the end.
```

- Step 7 (chunk 5 encoded):

```
02  01                              ; One data byte plus one trailing zero at the end.
```

- Step 8 (chunk 6 encoded):

```
02  28                              ; One data byte plus one trailing zero at the end.
```

Ping command (encoded):

```
01  EA  12  40  06  44  FD  A5  2A  9E  CC  18  D3  E2  FF  FE  E0  02  01  02  28
```

Ping command (encoded) – UART output:

```
00  01  EA  12  40  06  44  FD  A5  2A  9E  CC  18  D3  E2  FF  FE  02  01  02  28
```

^------ Start delimiter added to send the frame over UART.

# 2. KBI Frame Format

## 2.1. Frame format

Each UART frame transmitted to or received from an external host has the following format:

| HEADER | | | | | PAYLOAD |
|---|---|---|---|---|---|
| L0 | L1 | TYPE | CMD | CKS | ( Optional ≤ 1268 Bytes ) |

First there is a five bytes long header where:

- L0 : Most significant byte of the payload length.
- L1 : Less significant byte of the payload length.
- TYPE : Type descriptor.
- CMD : Command descriptor.
- CKS : Checksum byte. Is calculated by means of an XOR of all bytes of the frame.

Next to the header a variable length payload could be present as optional information for a specified command or response. The maximum length allowed for the payload is 1268 bytes. All the subfields contained in the payload must be in big endian.

**TYPE**, has the following bits:

| TYPE ( 1 Byte ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| FT | FT | FT | FT | FC | FC | FC | FC |

| | FT: **Frame Type** | | | | |
|---|---|---|---|---|---|
| FC: **Frame Code** | **0** Reserved | **1** **Command** | **2** **Response** | **3** **Notification** | **4** to **15** Reserved |
| 0 | - | Write / Execute | OK | Ping reply | - |
| 1 | - | Read | Value | Socket received data | - |
| 2 | - | Delete | Bad parameter | Named ping reply | - |
| 3 | - | - | Bad command | Named socket received data | - |
| 4 | - | - | Operation not allowed | Destination unreachable | - |
| 5 | - | - | Memory allocation error | - | - |
| 6 | - | - | Config. settings missing | - | - |
| 7 | - | - | Firmware update error | - | - |
| 8 | - | - | Busy | - | - |
| 9 to 15 | - | - | - | - | - |

## 2.2. Data representation

The payload of a command, response or notification may consist of one or more separate parameters, each with different representation. The main used types are:

- `HEX(n)`: generic hexadecimal value of variable length up to n bytes.
- `HEXN(n)`: generic hexadecimal value of fixed length of n bytes.
- `DEC(n)`: a decimal number (unsigned integer) represented in n bytes (fixed).
- `ENU`: specific case of DEC(1) in which only a defined enumeration of values is allowed (different for each command).
- `STR(a,b)`: ASCII string of arbitrary length between a and b characters. EOS is omitted for commands and it is included in responses or notifications.
- `STRN(n)`: ASCII string of fixed length of n characters (including EOS). More EOS bytes are allowed as padding when the string is shorter than required.
- `MAC`: specific case of HEXN(8) representing an 8-bytes long interface identifier (MAC address).
- `ADDR(n)`: specific case of HEXN(8) or HEXN(16) representing a 64-bits long IPv6 prefix or a 128-bits long IPv6 address.
- `LIST(x)`: used to indicate that the payload consists of a repetition of a certain pattern.

## 2.3. Commands and Responses

There are two ways of UART serial communications performed by KiNOS:

- One is the reception of a command from an external host and (after execution) its response. It means that every received command generates a response and it is transmitted back to the host.
- The other is the notification transmission from KiNOS device to an external host (detailed in the next chapter) to inform it of an asynchronous event.

A “command – wait response” sequence is needed by the external host side when a command is transmitted to KiNOS device. It means that the host has to wait for a response just after sending each command to device. It is recommended for the host side to have coherence between commands sent and responses received.

In some cases it could happen that the external host receives an asynchronous notification while it is waiting for a command response, so in that case it is choice of the host application to save it to be processed later or to reject it and continue waiting for the command response.

This example shows how “Command - wait response” sequence works:

| | |
|---|---|
| **Send command…** | **Host → KiNOS Stack** |
| Tx command - set channel 14 | `00 : 01 : 10 : 12 : 0d : 0e` |
| **Wait for response…** | **Host ← KiNOS Stack** |
| Rx response - OK | `00 : 00 : 20 : 12 : 32` |
| **Send next command…** | **Host → KiNOS Stack** |
| Tx command - get channel | `00 : 00 : 11 : 12 : 03` |
| **Wait for response…** | **Host ← KiNOS Stack** |
| Rx response - Value 14 | `00 : 01 : 21 : 12 : 3c : 0e` |

## 2.4. Notifications

A notification is performed by the KiNOS stack towards an external host to inform of an asynchronous event. For example a UDP to Serial traffic data reception which could be received at any time from the radio and transmitted over UART.

For a notification the CMD frame byte has no meaning.

| Ping reply | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Notification | | | |
| 30 | 00 | ADDR(16) · DEC(2) · DEC(2) · DEC(2) | Param. #1: <Remote IPv6 address><br>Param. #2: <Number of bytes><br>Param. #3: <Echo ID><br>Param. #4: <Echo sequence number> |

Example

KiNOS → Host

```
00 : 16 : 30 : 00 : 38 : fd : 00 : 0d : b8 : 00 : 00 : 00 : 00 : 00 : 00 : ff :
fe : 00 : 00 : 00 : 00 : 1c : ae : 55 : 72 : c2
```

Echo response received from fd00:db8::ff:fe00:0000 with 28 bytes, with ID 44629 and SN 29378.

| Socket receive | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Notification | | | |
| 31 | 00 | DEC(2) · DEC(2) · ADDR(16) · HEX(1232) | Param. #1: <Local port>, UDP destination port.<br>Param. #2: <Remote port>, UDP source port.<br>Param. #3: <Remote IPv6 address>, UDP source address.<br>Param. #4: <UDP payload> |

Example

KiNOS → Host

```
00 : 1a : 31 : 00 : 56 : ff : 98 : ff : 87 : fd : 6d : 18 : a4 : a2 : 50 : 0a : 26 :
18 : 7d : 57 : 20 : c7 : a3 : 8b : 96 : c9 : 01 : e2 : e6 : 37 : 00
```

udp rcv: saddr fd6d:18a4:a250:a26:187d:5720:c7a3:8b96 sport 65415 dport 65432 - 6 bytes

## Named ping reply

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| Notification | | | |
| 32 | 00 | STRN(32) · ADDR(16) · DEC(2) · DEC(2) · DEC(2) | Param. #1: <Remote domain name><br>Param. #2: <Remote IPv6 address><br>Param. #3: <Number of bytes><br>Param. #4: <Echo ID><br>Param. #5: <Echo sequence number> |

### Example

KiNOS → Host

```
00 : 36 : 32 : 00 : 3a : 6b : 69 : 72 : 61 : 6c : 65 : 2e : 63 : 6f : 6d : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 64 : ff : 9b : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 44 : 41 :
7a : ed : 00 : 08 : b4 : e5 : 46 : e4
```

ping reply: saddr 64:ff9b::4441:7aed [kirale.com] id 46309 sq 18148 - 8 bytes

## Named socket receive

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| Notification | | | |
| 33 | 00 | DEC(2) · DEC(2) · STRN(32) · ADDR(16) · HEX(1232) | Param. #1: <Local port>, UDP destination port.<br>Param. #2: <Remote port>, UDP source port.<br>Param. #3: <Remote domain name><br>Param. #4: <Remote IPv6 address>, UDP source address.<br>Param. #5: <UDP payload> |

### Example

KiNOS → Host

```
00 : ca : 33 : 00 : 41 : 04 : d2 : 16 : 33 : 63 : 6f : 61 : 70 : 2e : 6d : 65 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 64 : ff : 9b : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 86 : 66 : da : 12 : 64 : 45 : 2e : 6e : 00 : 00 : 68 : 1d : 48 : de : ea :
9a : 3e : 0e : da : c4 : 9b : 81 : 28 : b1 : 0b : ff : 3c : 2f : 74 : 65 : 73 : 74 :
3e : 3b : 72 : 74 : 3d : 22 : 74 : 65 : 73 : 74 : 22 : 3b : 63 : 74 : 3d : 30 : 2c :
3c : 2f : 76 : 61 : 6c : 69 : 64 : 61 : 74 : 65 : 3e : 3b : 72 : 74 : 3d : 22 : 76 :
61 : 6c : 69 : 64 : 61 : 74 : 65 : 22 : 3b : 63 : 74 : 3d : 30 : 2c : 3c : 2f : 68 :
65 : 6c : 6c : 6f : 3e : 3b : 72 : 74 : 3d : 22 : 54 : 79 : 70 : 65 : 31 : 22 : 3b :
63 : 74 : 3d : 30 : 3b : 69 : 66 : 3d : 22 : 49 : 66 : 31 : 22 : 2c : 3c : 2f : 62 :
6c : 25 : 43 : 33 : 25 : 41 : 35 : 62 : 25 : 43 : 33 : 25 : 41 : 36 : 72 : 73 : 79 :
6c : 74 : 65 : 74 : 25 : 43 : 33 : 25 : 42 : 38 : 79 : 3e : 3b : 72 : 74 : 3d : 22 :
62 : 6c : c3
```

udp rcv: saddr 64:ff9b::8666:da12 [coap.me] sport 5683 dport 1234 - 150 bytes

| Destination unreachable | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Notification | | | |
| 34 | 00 | ADDR(16) | Param. #1: <Remote IPv6 address> |

Example

KiNOS → Host

```
00 : 10 : 34 : 00 : 05 : fd : 1c : 75 : 68 : c2 : c1 : 00 : 00 : 00 : 00 : 00 : ff :
fe : 00 : dc : 03
```

dst unreachable: daddr fd1c:7568:c2c1::ff:fe00:dc03

# 3. Command Quick Reference Table

| # | Command | Operation | Parameters | Payload format | Default value |
|---|---|---|---|---|---|
| 0x00 | Clear | Execute | - | - | - |
| 0x01 | Thread version | Read | - | - | 2 (Thread v1.1) |
| 0x02 | Uptime | Read | - | - | - |
| 0x03 | Reset | Execute | - | - | - |
| 0x04 | Auto-join mode | Write (on) | - | - | 0 ('off') |
| | | Delete (off) | - | - | |
| | | Read | - | - | |
| 0x05 | Status | Read | - | - | none |
| 0x06 | Ping | Execute | <address> <nBytes> | ADDR(16) · DEC(2) | - |
| 0x07 | Ifdown | Execute | - | - | - |
| 0x08 | Ifup | Execute | - | - | - |
| 0x09 | Socket open/close | Write | (<localPort>) | DEC(2) | - |
| | | Delete | <localPort> | DEC(2) | |
| 0x0A | Software version | Read | - | - | FW version |
| 0x0B | Hardware version | Read | - | - | Factory set |
| 0x0C | Serial number | Read | - | - | Factory set |
| 0x0D | Extended MAC Address | Write | <extMAC> | MAC | Random |
| | | Read | - | - | |
| 0x0E | EUI-64 Address | Read | - | - | Factory set |
| 0x0F | Low power mode | Write (on) | - | - | 0 ('off') |
| | | Delete (off) | - | - | |
| | | Read | - | - | |
| 0x10 | Transmission power level | Write | <level> | ENU | 0 (+4 dBm) |
| | | Read | - | - | |
| 0x11 | PAN ID | Write | <panId> | HEXN(2) | - |
| | | Read | - | - | |

| # | Command | Operation | Parameters | Payload format | Default value |
|---|---|---|---|---|---|
| 0x12 | Channel | Write | <channel> | ENU | - |
| | | Read | - | - | |
| 0x13 | Ext. PAN ID | Write | <extPanId> | HEXN(8) | - |
| | | Read | - | - | |
| 0x14 | Network Name | Write | <netName> | STR(1,16) | - |
| | | Read | - | - | |
| 0x15 | Master key | Write | <mKey> | HEXN(16) | - |
| | | Read | - | - | |
| 0x16 | Commissioning Credential | Write | <pskC> | STR(6,255) | “THREAD” |
| | | Read | - | - | |
| 0x17 | Joiner Credential | Write | <pskD> | STR(6,32) | Factory set |
| | | Read | - | - | |
| 0x18 | Joiner management | Write | <eui64> <pskD> | MAC · STR(6,32) | - |
| | | Read | - | - | |
| | | Delete | (<eui64>) | (MAC) | |
| 0x19 | Role | Write | <role> | ENU | - |
| | | Read | - | - | |
| 0x1A | Short MAC Address | Read | - | - | - |
| 0x1B | Commissioner activation | Write (on) | - | - | off |
| | | Delete (off) | - | - | |
| 0x1C | Mesh local prefix | Write | <prefix> | ADDR(8) | - |
| | | Read | - | - | |
| 0x1D | Maximum number of children | Write | <maxChildren> | DEC(1) | 10 |
| | | Read | - | - | |
| 0x1E | Timeout | Write | <timeout> | DEC(4) | 240 |
| | | Read | - | - | |
| 0x1F | Ext. PAN ID Filter | Write | <ExtPanId> | HEXN(8) | - |
| | | Read | - | - | |
| | | Delete | - | - | |

| # | Command | Operation | Parameters | Payload format | Default value |
|---|---|---|---|---|---|
| 0x20 | IP address | Write | <ipv6 address> | ADDR(16) | - |
| | | Read | - | - | |
| | | Delete | <ipv6 address> | ADDR(16) | |
| 0x21 | Joiner port | Write | <joinerPort> | DEC(2) | 19786 |
| | | Read | - | - | |
| 0x22 | Hash EUI-64 Address | Read | - | - | Factory set |
| 0x23 | Polling rate | Write | <secs> | DEC(4) | 239 |
| | | Read | - | - | |
| 0x24 | Set out of band commissioning mode | Write | - | - | off |
| 0x25 | Steering data mode | Write | <mode> | ENU | on |
| 0x26 | Prefix | Write | <prefix> <len> <flags> | ADDR(8) · DEC(1) · HEXN(2) | - |
| | | Delete | <prefix> <len> | ADDR(8) · DEC(1) | |
| 0x27 | Route | Write | <route> <len> <flags> | ADDR(8) · DEC(1) · HEXN(2) | - |
| | | Delete | <route> <len> | ADDR(8) · DEC(1) | |
| 0x28 | Service | Write | <enterpriseId> <svcData> <srvData> | DEC(4) · HEX(249) · HEX(249) | - |
| | | Delete | <enterpriseId> <svcData> | DEC(4) · HEX(249) | |
| 0x29 | Parent information | Read | - | - | - |
| 0x2A | Router Table | Read | - | - | - |
| 0x2B | Leader Data | Read | - | - | - |
| 0x2C | Network Data | Read | - | - | - |
| 0x2D | Statistics | Read | - | - | - |
| 0x2E | Child Table | Read | - | - | - |
| 0x2F | Socket send | Execute | <localPort> <peerPort> <peerAddr> <content> | DEC(2) · DEC(2) · ADDR(16) · HEX(1232) | - |
| 0x30 | Firmware update | Execute | <Firmware block> | HEX(64) | - |
| 0x31 | Hardware mode | Write | <hwMode> | ENU | 1 |
| | | Read | - | - | |
| 0x32 | Led mode | Write (on) | - | - | 1 (‘on’) |
| | | Delete (off) | - | - | |

| # | Command | Operation | Parameters | Payload format | Default value |
|---|---|---|---|---|---|
| 0x33 | Vendor name | Write | <vendorName> | STR(0,32) | "Kirale Technologies" |
| | | Read | - | - | |
| 0x34 | Vendor model | Write | <vendorModel> | STR(0,32) | “KTWM102 module" |
| | | Read | - | - | |
| 0x35 | Vendor data | Write | <vendorData> | STR(0,64) | - |
| | | Read | - | - | |
| 0x36 | Vendor software version | Write | <vendorSwVersion> | STR(0,16) | "KiNOS v1.1" |
| | | Read | - | - | |
| 0x37 | Active timestamp | Write | <actTimestamp> | HEXN(8) | 0x0000000000010000 |
| | | Read | - | - | |
| 0x38 | Named ping | Execute | <domainName> <nBytes> | STRN(32) · DEC2) | - |
| 0x39 | Named socket send | Execute | <localPort> <peerPort> <domainName> <content> | DEC(2) · DEC(2) · STRN(32) · HEX(1232) | - |
| 0x3A | Services status | Read | - | - | - |
| 0x3B | Provisioning URL | Write | <provUrl> | STR(0,64) | empty |
| | | Read | - | - | |
| 0x3C | Commissioner Session ID | Read | - | - | - |
| 0x3D | MGMT_PENDING_GET.req | Execute | <peerAddr> <types> | ADDR(16) · HEX(292) | - |
| 0x3E | MGMT_PENDING_SET.req | Execute | <peerAddr> <tlvs> | ADDR(16) · HEX(292) | - |
| 0x3F | MGMT_ACTIVE_GET.req | Execute | <peerAddr> <types> | ADDR(16) · HEX(292) | - |
| 0x40 | MGMT_ACTIVE_SET.req | Execute | <peerAddr> <tlvs> | ADDR(16) · HEX(292) | - |
| 0x41 | MGMT_COMMISSIONER_GET.req | Execute | <peerAddr> <types> | ADDR(16) · HEX(292) | - |
| 0x42 | MGMT_COMMISSIONER_SET.req | Execute | <peerAddr> <tlvs> | ADDR(16) · HEX(292) | - |
| 0x43 | MGMT_PANID_QUERY.req | Execute | <channelMask> <panId> | HEXN(8) · HEXN(2) | - |

# 4. Detailed Command/Response Description

## Clear

Stop all running processes, clear the volatile memory and restore network configuration stored in non-volatile memory to the default settings. However there are some settings stored in non-volatile memory (transmission power, low power mode, hardware mode, auto-join mode, led mode, vendor software version, vendor model, vendor name, vendor data, and commissioning credential) which are not restored to default with clear execution.

This command may have a longer response delay than others. It is recommended to execute status command to assure that clear command has completed execution. Status has to be none when clear command has finished.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 00 | - | No parameters. |
| Responses | | | |
| 20 | 00 | - | OK. |
| 22 | 00 | - | Bad parameter: other than empty payload. |
| 24 | 00 | - | Operation not allowed. |

| Execution conditions |
|---|
| An execution of this command must finish before attempting any other command with a Frame Code different from Read, including the `clear` command itself. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 10 : 00 : 10` | `00 : 00 : 20 : 00 : 20` |
| Clear command execution. | Clear command executed successfully. |

# Thread Version

Show the KiNOS configured Thread version encoded as the MLE Version TLV.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 2 (Thread v1.1) | Permanent | No |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 01 | - | No parameters. |
| Responses | | | |
| 21 | 01 | DEC(2) | Param. #1: <Thread version> |
| 22 | 01 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 01 : 10` | `00 : 02 : 21 : 01 : 20 : 00 : 02` |
| Read the Thread version. | Thread version value: 2 (Thread v1.1). |

# Uptime

Show the time since the last power up in seconds, the UNIX Epoch time (only if there is an NTP service configured in the network) and the MCU temperature.

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 02 | - | No parameters. |
| Responses | | | |
| 21 | 02 | DEC(4) · DEC(4) · HEXN(1) | Param. #1: <Uptime>, seconds since last power up.<br>Param. #2: <Epoch>, UNIX Epoch time<br>Param. #3: <MCU temperature>, in ± degrees Celsius (one byte signed integer). |
| 22 | 02 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS<br>`00 : 00 : 11 : 02 : 13`<br>Read the uptime parameters. | KiNOS → Host<br>`00 : 09 : 21 : 02 : 70 : 00 : 01 : 21 : 64 : 00 : 00 : 00 : 00 : 1e`<br>Uptime: 0 days, 20 hours, 34 minutes and 44 seconds<br>Current UTC Time: 00:00:00 (NTP not available)<br>MCU Temperature: 30°C |

# Reset

Perform a software reset. All non-stored configuration will be lost.

This command is applied immediately. It is recommended to wait almost one second after receiving the OK response to continue sending other commands.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 03 | - | No parameters. |
| Responses | | | |
| 20 | 03 | - | OK. |
| 22 | 03 | - | Bad parameter: other than empty payload |

| Execution conditions |
|---|
| Always allowed. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 10 : 03 : 13` | `00 : 00 : 20 : 03 : 23` |
| Reset command execution. | Reset command executed successfully. |

# Auto-join Mode

Set the auto-join mode to ‘on’/’off’.

When auto-join mode is enabled and there was a successfully connection to a network (so the configuration is valid and it is already stored in non-volatile memory) the device will automatically start the joining to the same network in every power up.

Auto-join mode changes will only take effect after the next reset.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (off) | Immediate | No |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 04 | - | No parameters. |
| Responses | | | |
| 21 | 04 | ENU | Param. #1: <Auto-join mode>, 0 = ‘off’, 1 = ‘on’. |
| 22 | 04 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 04 : 15` | `00 : 01 : 21 : 04 : 24 : 00` |
| Read the auto-join mode. | Auto-join mode value: 0 (‘off’). |

| Write (on) | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 04 | - | Write with no parameters means set to ‘on’. |
| Responses | | | |
| 20 | 04 | - | OK. |
| 22 | 04 | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 00 : 10 : 04 : 14
```

Set the auto-join mode to ‘on’.

KiNOS → Host

```
00 : 00 : 20 : 04 : 24
```

Auto-join mode ‘on’ successfully stored.

## Delete (off)

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 04 | - | Delete with no parameters means set to ‘off’. |
| **Responses** | | | |
| 20 | 04 | - | OK. |
| 22 | 04 | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 00 : 12 : 04 : 16
```

Set the auto-join mode to ‘off’.

KiNOS → Host

```
00 : 00 : 20 : 04 : 24
```

Auto-join mode ‘off’ successfully stored.

# Status

Show the Thread connection status and the failure cause if the connection process failed.

| Value | | Status | Information | Description |
|---|---|---|---|---|
| MSB | LSB | | | |
| 0x00 | 0x00 | None | - | Not connected - not configured |
| 0x00 | 0x01 | None | Saved configuration | Not connected - configured |
| 0x00 | 0x02 | None | Network not found | Connection failed - network not found |
| 0x00 | 0x03 | None | Commissioning failed | Connection failed - commissioning error |
| 0x00 | 0x04 | None | Attaching failed | Connection failed - attaching error |
| 0x01 | -- | Booting | - | Booting the stack. |
| 0x02 | -- | Discovering | - | Scanning for networks |
| 0x03 | -- | Commissioning | - | Authentication in progress |
| 0x04 | -- | Attaching | - | Trying to connect to a network |
| 0x05 | -- | Joined | - | Connected |
| 0x06 | -- | Rebooting | - | Rebooting |
| 0x07 | -- | Changing partition | - | Migrating to another network |
| 0x10 | -- | Clearing | - | Clearing configuration |

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (none - not configured) | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 05 | - | No parameters. |
| **Responses** | | | |
| 21 | 05 | HEXN(2) | Param. #1: <Status>, according to the table. |
| 22 | 05 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 05 : 14
```

Show the status.

KiNOS → Host

```
00 : 02 : 21 : 05 : 27 : 00 : 01
```

Status is: none - saved configuration.

# Ping

Send an ICMPv6 echo request to a specified destination address with a specified payload length (random payload).

The ICMPv6 echo reply, if received, will be delivered to the host as an asynchronous notification. Depending on the role and the type of destination address, there could be a destination unreachable notification.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 06 | ADDR(16) · DEC(2) | Param #1: <Destination address><br>Param #2: <Payload size>, 0 to 1232. |
| Responses | | | |
| 20 | 06 | - | OK. |
| 22 | 06 | - | Bad parameter. |

**Execution conditions**

Status must be `joined`.

**Example**

Host → KiNOS

```
00 : 12 : 10 : 06 : a4 : 20 : 01 : 0d :
b8 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 77 : 00 : 03 : 00 : 40
```

Send an ICMPv6 echo request with 64 bytes payload size to the destination address 2001:db8::77:3.

KiNOS → Host

```
00 : 00 : 20 : 06 : 26
```

Command accepted.

# Ifdown

Shut down the Thread interface of the device. Clear not stored data and configuration and stop running processes. The module is disconnected from the network.

There might be a little delay between the execution of the command and the effective network shutdown. To make sure the shutdown process has finished the Status must be none.

If ifup command is executed after ifdown, the module resumes the connection to the network with the last stored configuration.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 07 | - | No parameters. |
| Responses | | | |
| 20 | 07 | - | OK. |
| 22 | 07 | - | Bad parameter. |

| Execution conditions |
|---|
| Always allowed. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 10 : 07 : 17` | `00 : 00 : 20 : 07 : 27` |
| Network shut down. | Command accepted. |

# Ifup

Start up the Thread interface of KiNOS stack. Starts the formation of a new network or the joining to an existing one depending on the configured role.

Role must be set before the execution of ifup command, otherwise a Configuration settings missing error will be generated.

Channel, PAN ID, network name and commissioning credential are optional settings and if not specified they will be automatically generated/selected.

If the out-of-band commissioning mode is set then channel, PAN ID, net name, local prefix, master key, extended PAN ID and commissioning credential must be configured before ifup execution, otherwise Configuration settings missing error will be generated.

Attaching to a Thread network may take a few seconds. The process will be finished when the status is joined or none.

Once the command is executed and the connection to a network is successful (status is joined), the module automatically stores the user configuration in non-volatile memory as a valid configuration.

When the command is executed while there is a valid network configuration stored, the module resumes the connection with that configuration. No modification of this configuration is allowed (a clear command needs to be executed in order to change any of the settings).

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 08 | - | No parameters. |
| Responses | | | |
| 20 | 08 | - | OK. |
| 22 | 08 | - | Bad parameter. |
| 26 | 08 | - | Configuration settings missing. |

### Execution conditions

Status must be none.

### Example

Host → KiNOS

```
00 : 00 : 10 : 08 : 18
```

Network start up.

KiNOS → Host

```
00 : 00 : 26 : 08 : 2e
```

Configuration settings missing.

# Socket Open/Close

Open/close a socket for UDP traffic via UART. The socket is bonded to a certain source port and to all the device's addresses. An additional command exists for sending traffic using this socket, and traffic received to this socket is delivered to the host via UART notifications.

All opened sockets are automatically deleted whenever the status stops being joined.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| Command | | | |
| 10 | 09 | [DEC(2)] | (Optional) Param. #1: <Source port> |
| Responses | | | |
| 21 | 09 | DEC(2) | Param. #1: <Source port>, randomly generated if not provided. |
| 22 | 09 | - | Bad parameter or specified port already in use. |

**Modification conditions**

Status must be `joined`.

**Example**

Host → KiNOS

```
00 : 02 : 10 : 09 : 12 : 30 : 39
```

Open a socket in the port 12345.

KiNOS → Host

```
00 : 02 : 21 : 09 : 23 : 30 : 39
```

Socket opened in the port 12345.

**Delete**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| Command | | | |
| 12 | 09 | DEC(2) | Param. #1: <Source port> |
| Responses | | | |
| 20 | 09 | - | OK. |
| 22 | 09 | - | Bad parameter or specified port not belonging to an open socket. |

### Modification conditions

Status must be `joined`.

### Example

Host → KiNOS

```
00 : 02 : 12 : 09 : 13 : 30 : 3a
```

Close a socket with source port 12346.

KiNOS → Host

```
00 : 00 : 22 : 09 : 2b
```

Bad parameter: no socket exists for the requested port.

# Software Version

Show the version strings for both the Bootloader and the KiNOS stack, separated by a “line feed” character and finishing with “end of string” character.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Depends on release | Yes | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| `11` | `0a` | `-` | No parameters. |
| **Responses** | | | |
| `21` | `0a` | `STR(0,256)` | Param. #1: <Software versions>, boot. ver. + LF + fw. ver. + EOS |
| `22` | `0a` | `-` | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 0a : 1b
```

Show the software version.

KiNOS → Host

```
00 : 32 : 21 : 0a : 40 : 42 : 6f : 6f :
74 : 6c : 6f : 61 : 64 : 65 : 72 : 20 :
76 : 31 : 2e : 31 : 2e : 36 : 38 : 30 :
33 : 2e : 36 : 35 : 33 : 36 : 39 : 0d :
0a : 4b : 69 : 4e : 4f : 53 : 20 : 76 :
31 : 2e : 32 : 2e : 37 : 32 : 35 : 37 :
2e : 36 : 33 : 37 : 39 : 34 : 00
```

Versions string: Bootloader v1.1.6803.65369 + LF + KiNOS v1.2.7257.63794 + EOS

# Hardware Version

Show the hardware version string of the device.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Factory set | Yes | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 0b | - | No parameters. |
| **Responses** | | | |
| 21 | 0b | STR(0,256) | Param. #1: <Hardware version> |
| 22 | 0b | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 0b : 1a
```

Show the hardware version.

KiNOS → Host

```
00 : 0a : 21 : 0b : 3b : 4b : 54 : 57 :
4d : 31 : 30 : 32 : 2d : 31 : 31
```

Version string: KTWM102-11

# Serial Number

Unique identification string of the device.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Factory set | Yes | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 0c | - | No parameters. |
| **Responses** | | | |
| 21 | 0c | STR(0,256) | Param. #1: <Serial number> |
| 22 | 0c | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 0c : 1d
```

Show the serial number string.

KiNOS → Host

```
00 : 23 : 21 : 0c : 64 : 4b : 54 : 57 :
4d : 31 : 30 : 32 : 2d : 31 : 31 : 2b :
32 : 30 : 31 : 38 : 30 : 31 : 2b : 38 :
34 : 30 : 34 : 44 : 32 : 30 : 30 : 30 :
30 : 30 : 30 : 30 : 30 : 30 : 35 : 00
```

Serial number string: KTWM102-11+201801+8404D20000000005

# Extended MAC Address

The MAC Extended Address is a random MAC generated on every device start up. It is used for internal communication processes when the device is commissioned on a network.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Random | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 0d | - | No parameters. |
| **Responses** | | | |
| 21 | 0d | MAC | Param. #1: <Extended MAC> |
| 22 | 0d | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 0d : 1c
```

Read the extended MAC address.

KiNOS → Host

```
00 : 08 : 21 : 0d : 57 : 7f : 0e : c0 :
f2 : f4 : 76 : 68 : da
```

Extended MAC address: 7f-0e-c0-f2-f4-76-68-da.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 0d | MAC | Param. #1: <Extended MAC> |
| **Responses** | | | |
| 20 | 0d | - | OK. |
| 22 | 0d | - | Bad parameter. |

### Modification conditions

Status must be none.

Example

Host → KiNOS

```
00 : 08 : 10 : 0d : 15 : aa : bb : cc :
dd : ee : ff : 00 : 11
```

Set the extended MAC address to aa-bb-cc-dd-ee-ff-00-11.

KiNOS → Host

```
00 : 00 : 20 : 0d : 2d
```

Extended MAC address successfully set.

# EUI-64 Address

Unique universally assigned IEEE Address for each device. IEEE assigned OUI for Kirale Technologies is 84-04-d2.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Factory set | Yes | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 0e | - | No parameters. |
| **Responses** | | | |
| 21 | 0e | MAC | Param. #1: <EUI-64> |
| 22 | 0e | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 0e : 1f
```

Read the EUI-64 address.

KiNOS → Host

```
00 : 08 : 21 : 0e : 70 : 84 : 04 : d2 :
00 : 00 : 00 : 00 : 05
```

EUI-64 address: 84-04-d2-00-00-00-00-05.

# Low Power Mode

Set the low consumption mode of the device to ‘on’/’off’. Any changes will only take effect after the next device reset.

When set to ‘on’ the device (radio and microprocessor) powers up in low consumption state, only awaked to attend UART commands (sleeping again after every response). The transmission power is set automatically to the level 11, but can be modified. After the `ifup` command is issued the device will only stay in low consumption state if the configured role was Sleepy End Device. In this case it will automatically awake exclusively when radio transmissions are required, or to attend UART commands. Also for SED case, if a LED is present it will not be turned on no matter the LED mode configuration.

When set to ‘off’ the device is always awake and its consumption depends on the radio transactions performed depending on the selected role.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (off) | Immediate | No |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 0f | - | No parameters. |
| Responses | | | |
| 21 | 0f | ENU | Param. #1: <Low power mode>, 0 = ‘off’, 1 = ‘on’. |
| 22 | 0f | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 0f : 1e` | `00 : 01 : 21 : 0f : 2e : 01` |
| Read the low power mode. | Low power mode value: 1 (‘on’). |

## Write (on)

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 0f | - | Write with no parameters means set to ‘on’. |
| **Responses** | | | |
| 20 | 0f | - | OK. |
| 22 | 0f | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 00 : 10 : 0f : 1f
```

Set the low power mode to ‘on’.

KiNOS → Host

```
00 : 00 : 20 : 04 : 24
```

Low power mode ‘on’ successfully stored.

## Delete (off)

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 0f | - | Delete with no parameters means set to ‘off’. |
| **Responses** | | | |
| 20 | 0f | - | OK. |
| 22 | 0f | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 00 : 12 : 0f : 1d
```

Set the low power mode to ‘off’.

KiNOS → Host

```
00 : 00 : 20 : 0f : 2f
```

Low power mode ‘off’ successfully stored.

# Transmission Power Level

Configuration of the transmission power level of the radio interface (delivered to the antenna), according to to the values in the following table:

| Value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tx power (dBm) | 4 | 3.7 | 3.4 | 3 | 2.5 | 2 | 1 | 0 | -1 | -2 | -3 | -4 | -6 | -8 | -12 | -17 |

Note: actual transmission power may vary depending on tolerances.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (+4 dBm) | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 10 | - | No parameters. |
| **Responses** | | | |
| 21 | 10 | ENU | Param. #1: <Transmission power> |
| 22 | 10 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 10 : 01
```

Read the transmission power.

KiNOS → Host

```
00 : 01 : 21 : 10 : 3b : 0b
```

Transmission power value: -4 dBm.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 10 | ENU | Param. #1: <Transmission power> |
| **Responses** | | | |
| 20 | 10 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 10 | - | Bad parameter: error in the parameter or out of range. |
| 24 | 10 | - | Operation not allowed. |

## Modification conditions

Status must be none.

## Example

Host → KiNOS

```
00 : 01 : 10 : 10 : 0a : 0b
```

Set the transmission power to -4 dBm.

KiNOS → Host

```
00 : 00 : 20 : 10 : 30
```

Transmission power stored successfully.

# PAN ID

Short identifier of the network.

If no PAN ID is specified and the node starts as “end device” (reed, fed, med, sed), network discovery will be performed on every channel without searching for a specific PAN ID. The connection process will be performed for every PAN ID found until one of them is successful.

If no PAN ID is specified and the node starts as “leader” the device will generate a random one and execute an active scan to verify that is not already used.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Random | When `joined` | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 11 | - | No parameters. |
| | | Responses | |
| 21 | 11 | HEXN(2) | Param. #1: <PAN ID> |
| 22 | 11 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 11 : 00` | `00 : 02 : 21 : 11 : 06 : fa : ce` |
| Read the PAN ID. | PAN ID value: 0xface. |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 11 | HEXN(2) | Param. #1: <PAN ID> |
| | | Responses | |
| 20 | 11 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 11 | - | Bad parameter: error in the parameter or out of range. |
| 24 | 11 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`.

## Example

Host → KiNOS

```
00 : 02 : 10 : 11 : 25 : 12 : 34
```

Set the PAN ID to 0x1234.

KiNOS → Host

```
00 : 00 : 24 : 11 : 35
```

Set PAN ID operation not allowed.

# Channel

Channel of the network. For Thread the channels 11 to 26 are defined.

If no channel is specified and the device starts as “end device” (reed, fed, med, sed), network discovery will be performed on every channel to find networks.

If no channel is specified and the device starts as “leader” an energy scan will be performed to pick the quietest one.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | When `joined` | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 12 | - | No parameters. |
| | | Responses | |
| 21 | 12 | ENU | Param. #1: <Channel> |
| 22 | 12 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 12 : 03` | `00 : 01 : 21 : 12 : 3d : 0f` |
| Read the Channel. | Channel value: 15. |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 12 | ENU | Param. #1: <Channel>, 11 to 26. |
| | | Responses | |
| 20 | 12 | - | OK. |
| 22 | 12 | - | Bad parameter. |

| | | | |
|---|---|---|---|
| 24 | 12 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`.

## Example

Host → KiNOS

```
00 : 01 : 10 : 12 : 0c : 0f
```

Set the Channel to 15.

KiNOS → Host

```
00 : 00 : 20 : 12 : 32
```

Channel stored successfully.

# Extended PAN ID

Extended network identifier for the network used to uniquely identify Thread networks in range.

It is required for out-of-band commissioning mode.

If not in out-of-band commissioning mode it will be random-generated at Thread interface start-up in case of Leader role, or acquired during attaching in case of other roles.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | When `joined` | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 13 | - | No parameters. |
| | | Responses | |
| 21 | 13 | HEXN(8) | Param. #1: <Extended PAN ID> |
| 22 | 13 | - | Bad parameter: other than empty payload. |

Example

Host → KiNOS

```
00 : 00 : 11 : 13 : 02
```

Read the Extended PAN ID.

KiNOS → Host

```
00 : 08 : 21 : 13 : 8f : 00 : 0d : b8 :
00 : 00 : 00 : 00 : 00
```

Extended PAN ID value: 0x000db80000000000.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 13 | HEXN(8) | Param. #1: <Extended PAN ID> |
| | | Responses | |
| 20 | 13 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 13 | - | Bad parameter. |
| 24 | 13 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`. Out-of-band commissioning mode must be active.

## Example

Host → KiNOS

```
00 : 08 : 10 : 13 : 83 : 11 : 22 : 33 :
44 : 55 : 66 : 77 : 88
```

Set the Extended PAN ID to 0x1122334455667788.

KiNOS → Host

```
00 : 00 : 24 : 13 : 37
```

Set Extended PAN ID operation not allowed.

# Network Name

Human-readable name of the network.

It is required for out-of-band commissioning mode.

If not in out-of-band commissioning mode it will be random-generated at Thread interface start-up using kite_xxxx format in case of “leader”, or acquired during attaching in case of other roles.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | When `joined` | Yes |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 14 | - | No parameters. |
| Responses | | | |
| 21 | 14 | STR(1,17) | Param. #1: <Network name> |
| 22 | 14 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 14 : 05` | `00 : 0a : 21 : 14 : 55 : 4d : 79 : 4e : 65 : 74 : 77 : 6f : 72 : 6b : 00` |
| Read the Network name. | Network name value: “MyNetwork”. |

| Write | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 14 | STR(1,16) | Param. #1: <Network name> |
| Responses | | | |
| 20 | 14 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 14 | - | Bad parameter. |
| 24 | 14 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`. Out-of-band commissioning mode must be active.

## Example

Host → KiNOS

```
00 : 09 : 10 : 14 : 67 : 4d : 79 : 4e :
65 : 74 : 77 : 6f : 72 : 6b
```

Set the Network name to “MyNetwork”.

KiNOS → Host

```
00 : 00 : 20 : 14 : 34
```

Network name stored successfully.

# Master Key

Shared secret master key from which network security keys will be derived in case of out of band commissioning.

It is required for out-of-band commissioning mode.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | When `joined` | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 15 | - | No parameters. |
| | | Responses | |
| 21 | 15 | HEXN(16) | Param. #1: <Master key> |
| 22 | 15 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 15 : 04
```

Read the Master key.

KiNOS → Host

```
00 : 10 : 21 : 15 : 24 : 00 : 11 : 22 :
33 : 44 : 55 : 66 : 77 : 88 : 99 : aa :
bb : cc : dd : ee : ff
```

Master key value:

0x00112233445566778899aabbccddeeff.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 15 | HEXN(16) | Param. #1: <Master key> |
| | | Responses | |
| 20 | 15 | - | OK. |
| 22 | 15 | - | Bad parameter. |

| | | | |
|---|---|---|---|
| 24 | 15 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`. Out-of-band commissioning mode must be active.

## Example

Host → KiNOS

```
00 : 10 : 10 : 15 : 15 : 00 : 11 : 22 :
33 : 44 : 55 : 66 : 77 : 88 : 99 : aa :
bb : cc : dd : ee : ff
```

Set the Master key to

0x00112233445566778899aabbccddeeff.

KiNOS → Host

```
00 : 00 : 20 : 15 : 35
```

Master key stored successfully.

# Commissioning Credential

A human-scaled pass-phrase for use in authenticating that a device may petition to become the Commissioner of the network.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| THREAD | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 16 | - | No parameters. |
| **Responses** | | | |
| 21 | 16 | STR(7,256) | Param. #1: <Commissioning credential> |
| 22 | 16 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 16 : 07
```

Read the Commissioning credential.

KiNOS → Host

```
00 : 0b : 21 : 16 : 37 : 4d : 79 : 50 :
61 : 73 : 73 : 77 : 6f : 72 : 64 : 00
```

Commissioning credential value: “MyPassword”.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 16 | STR(6,255) | Param. #1: <Commissioning credential> |
| **Responses** | | | |
| 20 | 16 | - | OK. |
| 22 | 16 | - | Bad parameter. |
| 24 | 16 | - | Operation not allowed. |

### Modification conditions

Status must be none, except none `- saved configuration`.

Example

Host → KiNOS

```
00 : 0a : 10 : 16 : 07 : 4d : 79 : 50 :
61 : 73 : 73 : 77 : 6f : 72 : 64
```

Set the Commissioning credential to “MyPassword”.

KiNOS → Host

```
00 : 00 : 20 : 16 : 36
```

Commissioning credential stored successfully.

# Joiner Credential

A human-scaled pass-phrase for use in authenticating that a new Joiner device is the correct one and proceed on the commissioning.

Allowed characters are ‘0’-’9’ and ‘A’-’Y’ excluding ‘I’, ‘O’, ‘Q’.

The default value is associated to the Serial number of the device.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Factory set | Immediate | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 17 | - | No parameters. |
| | | Responses | |
| 21 | 17 | STR(7,33) | Param. #1: <Commissioning credential> |
| 22 | 17 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 17 : 06` | `00 : 11 : 21 : 17 : 50 : 38 : 34 : 30 : 34 : 44 : 32 : 30 : 30 : 30 : 30 : 30 : 30 : 30 : 30 : 30 : 39 : 00` |
| Read the Joiner credential. | Joiner credential value: “8404D20000000009”. |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 17 | STR(6,32) | Param. #1: <Joiner credential> |
| | | Responses | |
| 20 | 17 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 17 | - | Bad parameter: wrong length or invalid characters. |
| 24 | 17 | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`.

## Example

Host → KiNOS

```
00 : 07 : 10 : 17 : 6d : 61 : 62 : 63 :
49 : 4f : 51 : 5a
```

Set the Joiner credential to “abcIOQZ”.

KiNOS → Host

```
00 : 00 : 22 : 17 : 35
```

Bad parameter for Joiner credential configuration.

# Joiner Management

Manages joiners in the list of allowed joiner devices of the Leader, up to a total of 10 devices.

This setting is only distributed to the leader (to upgrade the Network Data) if the node in which this setting is applied is acting as the network commissioner.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 18 | - | No parameters. |
| Responses | | | |
| 21 | 18 | [LIST(MAC)] | List element param. #1: <Joiner EUI-64> |
| 22 | 18 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 18 : 09` | `00 : 10 : 21 : 18 : 6a : 11 : 22 : 33 : 44 : 55 : 66 : 77 : 88 : 84 : 04 : d2 : 00 : 00 : 00 : 00 : 99` |
| Read the allowed joiners list. | Allowed joiners list: [11-22-33-44-55-66-77-88, 84-04-d2-00-00-00-00-99] . |

| Write | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 18 | MAC · STR(6,32) | Param. #1: <Joiner EUI-64><br>Param. #2: <Joiner credential> |
| Responses | | | |
| 20 | 18 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 18 | - | Bad parameter. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 0e : 10 : 18 : 80 : 11 : 22 : 33 :
44 : 55 : 66 : 77 : 88 : 54 : 48 : 52 :
45 : 41 : 44
```

Add a joiner with EUI-64 11-22-33-44-55-66-77-88 and joiner credential “THREAD”.

KiNOS → Host

```
00 : 00 : 20 : 18 : 38
```

Joiner successfully stored.

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 18 | [MAC] | (Optional) Param. #1: <Joiner EUI-64><br>With no parameters the full list is deleted. |
| **Responses** | | | |
| 20 | 18 | - | OK. |
| 22 | 18 | - | Bad parameter. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 08 : 12 : 18 : 8b : 11 : 22 : 33 :
44 : 55 : 66 : 77 : 89
```

Remove the entry with the EUI-64 11-22-33-44-55-66-77-89 from the list of allowed joiners.

KiNOS → Host

```
00 : 00 : 20 : 18 : 38
```

Removal operation accepted (no error when entry does not exist).

# Role

The desired role of the device in the network.

| Value | Role |
|---|---|
| 0 | Not configured |
| 1 | Router |
| 2 | Router eligible end device |
| 3 | Full end device |
| 4 | Minimal end device |
| 5 | Sleepy end device |
| 6 | Leader |

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (Not configured) | When `joined` | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 19 | - | No parameters. |
| **Responses** | | | |
| 21 | 19 | ENU | Param. #1: <Role>, 1 to 6. |
| 22 | 19 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 19 : 08
```

Read the role.

KiNOS → Host

```
00 : 01 : 21 : 19 : 39 : 00
```

Role not configured.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 19 | ENU | Param. #1: <Role>, 1 to 6. |
| **Responses** | | | |

| | | | |
|---|---|---|---|
| 20 | 19 | - | OK. |
| 22 | 19 | - | Bad parameter. |

## Modification conditions

Status must be none, except `none - saved configuration`. Also allowed when status is `joined` to trigger some role transition mechanisms (SED → MED, SED → FED, MED→ FED).

## Example

Host → KiNOS

```
00 : 01 : 10 : 19 : 0b : 03
```

Configure the role as FED.

KiNOS → Host

```
00 : 00 : 20 : 19 : 39
```

Role successfully stored.

# Short MAC Address

Short MAC Address acquired when the device is connected to a network. Also known as RLOC16 address.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0xFFFF | When `joined` | Yes |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 1a | - | No parameters. |
| Responses | | | |
| 21 | 1a | HEXN(2) | Param. #1: <RLOC16> |
| 22 | 1a | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 1a : 0b` | `00 : 02 : 21 : 1a : 39 : 00 : 00` |
| Read the RLOC16. | RLOC16 value: 0x0000. |

# Commissioner Activation

Enable or disable the commissioner functionality in a device.

Only one active commissioner at a time is allowed in a Thread network. Even after a successful enabling command, it might happen that the Leader rejects the device's commissioner petition.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0 (off) | No | Yes |

**Write (on)**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| `10` | `1b` | - | Write with no parameters means set to 'on'. |
| **Responses** | | | |
| `20` | `1b` | - | OK. |
| `22` | `1b` | - | Bad parameter: other than empty payload. |

**Modification conditions**

Always allowed.

**Example**

Host → KiNOS

```
00 : 00 : 10 : 1b : 0b
```

Set the commissioner mode to 'on'.

KiNOS → Host

```
00 : 00 : 20 : 1b : 3b
```

Commissioner mode 'on' successfully stored.

**Delete (off)**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| `12` | `1b` | - | Delete with no parameters means set to 'off'. |
| **Responses** | | | |

| | | | |
|---|---|---|---|
| 20 | 1b | - | OK. |
| 22 | 1b | - | Bad parameter: other than empty payload. |

## Modification conditions

Always allowed.

## Example

Host → KiNOS

```
00 : 00 : 12 : 1b : 09
```

Set the commissioner mode to ‘off’.

KiNOS → Host

```
00 : 00 : 20 : 1b : 3b
```

Commissioner mode ‘off’ successfully stored.

# Mesh Local Prefix

The IPv6 prefix used for on-mesh internal communication between devices in the same network.

It is required for out-of-band commissioning mode.

If not in out-of-band commissioning mode it will be random-generated at Thread interface start-up in case of Leader role, or acquired during attaching in case of other roles.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | When `joined` | yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 1c | - | No parameters. |
| **Responses** | | | |
| 21 | 1c | ADDR(8) | Param. #1: <Mesh local prefix> |
| 22 | 1c | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 1c : 0d
```

Read the mesh local prefix.

KiNOS → Host

```
00 : 08 : 21 : 1c : 7d : fd : 00 : 0d :
b8 : 00 : 00 : 00 : 00
```

Mesh local prefix value: fd00:db8::.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 1c | ADDR(8) | Param. #1: <Mesh local prefix> |
| **Responses** | | | |
| 20 | 1c | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 1c | - | Bad parameter. |
| 24 | 1c | - | Operation not allowed. |

## Modification conditions

Status must be none, except `none - saved configuration`. Out-of-band commissioning mode must be active.

## Example

Host → KiNOS

```
00 : 08 : 10 : 1c : 89 : fd : 12 : 34 :
56 : 00 : 00 : 00 : 00
```

Set the mesh local prefix to fd12:3456::

KiNOS → Host

```
00 : 00 : 24 : 1c : 38
```

Set mesh local prefix operation not allowed.

# Maximum Number of Children

Maximum number of children that can be attached to the device.

This setting has only effect when the device behaves as router in the network.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 10 | When `joined` | yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 1d | - | No parameters. |
| **Responses** | | | |
| 21 | 1d | DEC(1) | Param. #1: <Max. children>, 10 to 64. |
| 22 | 1d | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

`00 : 00 : 11 : 1d : 0c`

Read the maximum number of children.

KiNOS → Host

`00 : 01 : 21 : 1d : 37 : 0a`

Maximum number of children value: 10.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 1d | DEC(1) | Param. #1: <Max. children>, 10 to 64. |
| **Responses** | | | |
| 20 | 1d | - | OK. |
| 22 | 1d | - | Bad parameter. |
| 24 | 1d | - | Operation not allowed. |

### Modification conditions

Status must be none, except `none - saved configuration`.

Example

Host → KiNOS

```
00 : 01 : 10 : 1d : 6c : 60
```

Set the maximum number of children to 96.

KiNOS → Host

```
00 : 00 : 22 : 1d : 3f
```

Bad parameter.

# Child Timeout

Time in seconds within an end device MUST check-in with its parent with a keep alive request otherwise the parent considers this child is out of the network and proceeds to remove it from its children table.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 240 seconds | When `joined` | yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 1e | - | No parameters. |
| | | Responses | |
| 21 | 1e | DEC(4) | Param. #1: <Child timeout>, 2 to 4147199. |
| 22 | 1e | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 1e : 0f
```

Read the child timeout.

KiNOS → Host

```
00 : 04 : 21 : 1e : cb : 00 : 00 : 00 : f0
```

Child timeout value: 240.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 1e | DEC(4) | Param. #1: <Child timeout>, 2 to 4147199. |
| | | Responses | |
| 20 | 1e | - | OK. |
| 22 | 1e | - | Bad parameter. |
| 24 | 1e | - | Operation not allowed. |

### Modification conditions

Status must be none.

### Example

Host → KiNOS

```
00 : 04 : 10 : 1e : 14 : 00 : 00 : 0e : 10
```

Set the child timeout to 3600.

KiNOS → Host

```
00 : 00 : 20 : 1e : 3e
```

Child timeout value accepted successfully.

# Extended Pan Id Filter

A filter list to specify some known networks to which connection is not desired and will be discarded during network discovery.

This list is limited to a maximum of ten items.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 1f | - | No parameters. |
| **Responses** | | | |
| 21 | 1f | [LIST(HEXN(8))] | List element param. #1: <Extended PAN ID> |
| 22 | 1f | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 1f : 0e
```

Read the extended PAN ID filter list.

KiNOS → Host

```
00 : 00 : 20 : 1f : 3f
```

Empty extended PAN ID filter list.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 1f | HEXN(8) | Param. #1: <Extended PAN ID>, add to the list. |
| **Responses** | | | |
| 20 | 1f | - | OK. |
| 22 | 1f | - | Bad parameter. |

### Modification conditions

Status must be none.

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 08 : 10 : 1f : 07 : 12 : 34 : 56 : 78 : 12 : 34 : 56 : 78` | `00 : 00 : 20 : 1f : 3f` |
| Add extended PAN ID to the list. | Extended PAN ID successfully added to the list. |

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 1f | - | No parameters, delete the full list. |
| **Responses** | | | |
| 20 | 1f | - | OK. |
| 22 | 1f | - | Bad parameter. |

### Modification conditions

Always allowed.

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 12 : 1f : 0d` | `00 : 00 : 20 : 1f : 3f` |
| Clear the extended PAN ID filter list. | Clear operation accepted. |

# IP Address

The list of unicast and multicast IPv6 addresses of the Thread interface of the device.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| System and auto-generated addresses | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 20 | - | No parameters. |
| **Responses** | | | |
| 21 | 20 | [LIST(ENU · ADDR(16))] | List element param. #1: <State> 0 = tentative, 1 = registered, 4 = invalid.<br>List element param. #2: <IPv6 address> |
| 22 | 20 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 20 : 31
```

Read the IPv6 addresses list.

KiNOS → Host

```
00 : 99 : 21 : 20 : 06 : 01 : fe : 80 :
00 : 00 : 00 : 00 : 00 : 00 : ac : 79 :
50 : 2d : 84 : b1 : a8 : 05 : 01 : fd :
00 : 0d : b8 : 00 : 00 : 00 : 00 : 41 :
34 : bd : a0 : e9 : 24 : 17 : 26 : 01 :
fd : 00 : 0d : b8 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : ff : fe : 00 : 00 : 01 :
04 : ab : cd : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 12 :
34 : 01 : ff : 02 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 01 : 01 : ff : 03 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 01 : 01 : ff : 33 : 00 : 40 :
fd : 00 : 0d : b8 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 01 : 01 : ff : 32 : 00 :
40 : fd : 00 : 0d : b8 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 01 : 01 : ff : 03 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : fc
```

IPv6 addresses list:

[R] fe80::ac79:502d:84b1:a805
[R] fd00:db8::4134:bda0:e924:1726
[R] fd00:db8::ff:fe00:1
[I] abcd::1234
[R] ff02::1
[R] ff03::1
[R] ff33:40:fd00:db8::1
[R] ff32:40:fd00:db8::1
[R] ff03::fc

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 20 | ADDR(16) | Param. #1: <IPv6 addresses> |
| **Responses** | | | |
| 20 | 20 | - | OK. |
| 22 | 20 | - | Bad parameter. |

### Modification conditions

Status must be `joined`.

### Example

Host → KiNOS

```
00 : 10 : 10 : 20 : 60 : ab : cd : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 12 : 34
```

Configure the address abcd::1234.

KiNOS → Host

```
00 : 00 : 20 : 20 : 00
```

Address configured successfully.

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 20 | ADDR(16) | Param. #1: <IPv6 address> |
| **Responses** | | | |
| 20 | 20 | - | OK. |
| 22 | 20 | - | Bad parameter. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 10 : 12 : 20 : 63 : ab : cd : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 12 : 35
```

Remove the address address abcd::1235.

KiNOS → Host

```
00 : 00 : 20 : 20 : 00
```

Removal operation accepted.

# Joiner Port

UDP port used by the Joiner Router.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 49786 | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 21 | - | No parameters. |
| **Responses** | | | |
| 21 | 21 | DEC(2) | Param. #1: <Joiner port> |
| 22 | 21 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 21 : 30
```

Read the joiner port.

KiNOS → Host

```
00 : 02 : 21 : 21 : ba : c2 : 7a
```

Joiner port value: 49786.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 21 | DEC(2) | Param. #1: <Joiner port> |
| **Responses** | | | |
| 20 | 21 | - | OK. |
| 22 | 21 | - | Bad parameter. |
| 24 | 21 | - | Operation not allowed. |

### Modification conditions

Status must be none.

Example

Host → KiNOS

```
00 : 02 : 10 : 21 : 54 : ff : 98
```

Set the joiner port to 65432.

KiNOS → Host

```
00 : 00 : 20 : 21 : 01
```

Joiner port value stored successfully.

# EUI-64 Address Hash

Hash value of the Factory EUI-64 using SHA256 algorithm.

Hash EUI-64 Address is used for commissioning packet transaction. It could be useful to identify the owner of some radio packets before it becomes part of the network.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Factory set | Yes | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 22 | - | No parameters. |
| **Responses** | | | |
| 21 | 22 | MAC | Param. #1: <Hash EUI-64> |
| 22 | 22 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 22 : 33
```

Read the EUI-64 address hash.

KiNOS → Host

```
00 : 08 : 21 : 22 : b6 : 33 : 8a : 7b :
1a : 21 : 7b : 42 : 7d
```

EUI-64 address hash: 33-8a-7b-1a-21-7b-42-7d.

# Polling Rate

Period in seconds for which a SED device will generate MAC Data Request frames for its parent.

Whenever a child timeout smaller than the polling rate is configured, the polling rate automatically adopts the value (Child timeout -1).

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 239 seconds | When `joined` | Yes |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 23 | - | No parameters. |
| Responses | | | |
| 21 | 23 | DEC(4) | Param. #1: <Polling rate>, 1 to 4147198. |
| 22 | 23 | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 00 : 11 : 23 : 32` | `00 : 04 : 21 : 23 : 05 : 00 : 00 : 00 : 03` |
| Read the polling rate. | Polling rate value: 3. |

| Write | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 23 | DEC(4) | Param. #1: <Polling rate>, 1 to 4147198. |
| Responses | | | |
| 20 | 23 | - | OK. |
| 22 | 23 | - | Bad parameter. |
| 24 | 23 | - | Operation not allowed. |

## Modification conditions

Status must be none.

## Example

Host → KiNOS

```
00 : 04 : 10 : 23 : 37 : 00 : 00 : 00 : 00
```

Set the polling rate to 0.

KiNOS → Host

```
00 : 00 : 22 : 23 : 01
```

Bad parameter.

# Out-of-Band Commissioning Mode

Switch to out-of-band commissioning mode.

This mode allows the device to be commissioned by the user by configuring it with all the necessary parameters and credentials to join a specific network without following the in band commissioning process.

If out-of-band commissioning mode is set the following parameters are required before issuing `ifup`: Role, Channel, PAN ID, Network name, Local prefix, Master key, Extended PAN ID and Commissioning Credential.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Off | When `joined` | Yes |

**Execute**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 24 | - | No parameters. |
| **Responses** | | | |
| 20 | 24 | - | OK. |
| 22 | 24 | - | Bad parameter. |
| 24 | 24 | - | Operation not allowed. |

**Modification conditions**

Status must be none.

**Example**

Host → KiNOS

```
00 : 00 : 10 : 24 : 34
```

Switch to out-of-band mode.

KiNOS → Host

```
00 : 00 : 20 : 24 : 04
```

Device switched to out-of-band mode.

# Steering Data Mode

Steering data filter indicates which Joiners are eligible to join the Thread Network. This data is published in the network by the active Commissioner. It is possible to set a steering data with all joiners allowed, all joiners blocked or specific joiners allowed (filter activated). Joiners may only attempt to join on whenever they match the steering data filter.

| Value | Mode |
|---|---|
| 0 | All nodes allowed |
| 1 | All nodes blocked |
| 2 | Steering data filter activated |

If steering data filter is activated the joiners which will be allowed to join the network will be only those configured with “Joiner management” command in the Commissioner joiners list.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 2 | No | Yes |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 25 | ENU | Param. #1: <Steering data mode> |
| | | Responses | |
| 20 | 25 | - | OK. |
| 22 | 25 | - | Bad parameter. |
| 24 | 25 | - | Operation not allowed. |

**Modification conditions**

Device must be Commissioner.

**Example**

Host → KiNOS

```
00 : 01 : 10 : 25 : 34 : 00
```

Set the steering data mode to All nodes allowed.

KiNOS → Host

```
00 : 00 : 24 : 25 : 01
```

Operation not allowed.

# Prefix

Handling of prefixes announced by the device to the Thread Network.

Prefix flags are encoded the same way as in the Thread Border Router TLV, with the exception of the LSB which is used to mark it as stable when set:

| MSB | | | | | | | | | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Prf | Prf | P | S | D | C | R | O | N | - | - | - | - | - | - | Stb |

The list of available prefixes on the Thread Network is obtained with the Network Data command.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 26 | ADDR(16) · DEC(1) · HEXN(2) | Param. #1: <Prefix><br>Param. #2: <Length><br>Param. #3: <Flags> |
| **Responses** | | | |
| 20 | 26 | - | OK. |
| 22 | 26 | - | Bad parameter. |
| 24 | 26 | - | Operation not allowed. |

**Modification conditions**

Status must be `joined` and role must be other than `med` or `sed`.

**Example**

Host → KiNOS

```
00 : 13 : 10 : 26 : 60 : ca : fe : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 40 : 30 : 01
```

KiNOS → Host

```
00 : 00 : 20 : 26 : 06
```

| | |
|---|---|
| Add the stable preferred SLAAC prefix cafe:: of length 64. | Prefix add command successfully accepted. |

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 26 | ADDR(16) · DEC(1) | Param. #1: <Prefix><br>Param. #2: <Length> |
| **Responses** | | | |
| 20 | 26 | - | OK. |
| 22 | 26 | - | Bad parameter. |
| 24 | 26 | - | Operation not allowed. |

### Modification conditions

Status must be `joined`.

### Example

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 11 : 12 : 26 : 03 : ab : cd : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 40` | `00 : 00 : 24 : 26 : 02` |
| Remove the prefix abcd::/64. | Removal operation not allowed. |

# Route

Handling of routes to external prefixes announced by the device to the Thread Network.

The route preference (0 = medium, 1 = low, 3 = high) and stable condition are encoded same way as for the prefix command.

| MSB | | | | | | | | | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Prf | Prf | - | - | - | - | - | - | - | - | - | - | - | - | - | Stb |

The list of available routes on the Thread Network is obtained with the Network Data command.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 27 | ADDR(16) · DEC(1) · HEXN(2) | Param. #1: <Prefix><br>Param. #2: <Length><br>Param. #3: <Flags> |
| **Responses** | | | |
| 20 | 27 | - | OK. |
| 22 | 27 | - | Bad parameter. |
| 24 | 27 | - | Operation not allowed. |

**Modification conditions**

Status must be `joined` and role must be other than `med` or `sed`.

**Example**

| Host → KiNOS | KiNOS → Host |
|---|---|

| | |
|---|---|
| `00 : 13 : 10 : 27 : 45 : 00 : 64 : ff : 9b : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 60 : 00 : 01`<br>Add a stable route to 64:ff9b::/96 with medium preference. | `00 : 00 : 20 : 27 : 07`<br>Route add command successfully accepted. |

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 27 | ADDR(16) · DEC(1) | Param. #1: <Route><br>Param. #2: <Length> |
| **Responses** | | | |
| 20 | 27 | - | OK. |
| 22 | 27 | - | Bad parameter. |
| 24 | 27 | - | Operation not allowed. |

### Modification conditions

Status must be `joined`.

### Example

| | |
|---|---|
| Host → KiNOS<br>`00 : 11 : 12 : 27 : 44 : 00 : 64 : ff : 9b : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 60`<br>Remove the route to 64:ff9b::/96. | KiNOS → Host<br>`00 : 00 : 20 : 27 : 07`<br>Route remove command successfully accepted. |

# Service

Handling of services announced by the device to the Thread Network.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 28 | HEXN(4) · DEC(1) · HEX(250) · DEC(1) · HEX(250) | Param. #1: <Enterprise ID><br>Param. #2: <Service data length><br>Param. #3: <Service data><br>Param. #4: <Server data length><br>Param. #5: <Server data> |
| **Responses** | | | |
| 20 | 28 | - | OK. |
| 22 | 28 | - | Bad parameter. |
| 24 | 28 | - | Operation not allowed. |

**Modification conditions**

Status must be `joined`.

**Example**

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 0f : 10 : 28 : e9 : 00 : 00 : c0 : 0e : 05 : 11 : 22 : 33 : 44 : 55 : 04 : aa : bb : cc : dd` | `00 : 00 : 20 : 28 : 08` |
| Add a service with the Kirale's Private Enterprise Number (49166), service data 0x1122334455 and server data 0xaabbccdd | Service add command successfully accepted. |

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 28 | HEXN(4) · DEC(1) · HEX(250) | Param. #1: <Enterprise ID><br>Param. #2: <Service data length><br>Param. #3: <Service data> |
| **Responses** | | | |
| 20 | 28 | - | OK. |
| 22 | 28 | - | Bad parameter. |
| 24 | 28 | - | Operation not allowed. |

### Modification conditions

Status must be `joined`.

### Example

Host → KiNOS

```
00 : 09 : 12 : 28 : 27 : 00 : 00 : af :
aa : 11 : 22 : 33 : 44 : 55
```

Remove the service with the Kirale's Private Enterprise Number (49166) and service data 0x1122334455.

KiNOS → Host

```
00 : 00 : 20 : 28 : 08
```

Service remove command successfully accepted.

# Parent Information

Show the short and extended MAC addresses of the Thread parent of the device.

If the device is acting as a Router this information will be empty.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 29 | - | No parameters. |
| **Responses** | | | |
| 21 | 29 | [MAC · HEXN(2)] | Param. #1: <Parent extended MAC address><br>Param. #2: <Parent short MAC address> |
| 22 | 29 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 29 : 38
```

Read the parent information.

KiNOS → Host

```
00 : 0a : 21 : 29 : 00 : e6 : e5 : b0 :
21 : 0c : 2d : 58 : e9 : 00 : 00
```

Parent ext. MAC address: e6-e5-b0-21-0c-2d-58-e9

Parent short MAC address: 0x0000

# Router Table

Show a list of best routes (next hop) to reach every active router of the network and the path cost.

If device doesn't have any router in the router table the response is empty.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 2a | - | No parameters. |
| **Responses** | | | |
| 21 | 2a | [LIST(HEXN(2) · ENU · ENU · ENU · DEC(1))] | List elem. param. #1: <Target router short MAC addr.><br>List elem. param. #2: <Target Router ID>, 0 to 62.<br>List elem. param. #3: <Next hop Router ID>, 0 to 62.<br>List elem. param. #4: <Route cost>, 0 = unreachable, 1 to 15 = best to worst.<br>List elem. param. #5: <Link margin>, in dB or 0 for self Router ID. |
| 22 | 2a | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 2a : 3b
```

Read the router table.

KiNOS → Host

```
00 : 0c : 21 : 2a : 07 : d4 : 00 : 35 :
35 : 01 : 00 : 80 : 00 : 20 : 35 : 01 :
41
```

Router table:

```
+-----------------------------------------+
| RLOC16 | R. ID | N.H. ID | Cost |  LM  |
+--------+-------+---------+------+------+
| 0xd400 |  53   |   53    |  01  |  00  |
| 0x8000 |  32   |   53    |  01  |  65  |
+-----------------------------------------+
```

# Leader Data

Read the available Thread Network Leader Data from the device.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 2b | - | No parameters. |
| **Responses** | | | |
| 21 | 2b | [LIST(HEXN(4) · DEC(1)· DEC(1)· DEC(1) · DEC(1))] | List elem. param. #1: <Partition ID><br>List elem. param. #2: <Weighting ><br>List elem. param. #3: <Data version><br>List elem. param. #4: <Stable data version><br>List elem. param. #5: <Leader Router ID> |
| 22 | 2b | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 2b : 3a
```

Read the leader data.

KiNOS → Host

```
00 : 08 : 21 : 2b : 18 : 15 : fa : d4 :
81 : 40 : 17 : c2 : 35
```

Leader data:

- Partition ID: 0x15fad481
- Weighting: 64
- Data version: 23
- Stable data version: 194
- Leader Router ID: 53

# Network Data

Read the available Network Data from the device.

Al device roles use both the stable and unstable Network Data, except `sed` that only uses the stable one.

The returned payload is encoded as specified by Thread Specification, section *5.18 Network Data Encoding*.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

| Read | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 11 | 2c | - | No parameters. |
| Responses | | | |
| 21 | 2c | [HEX(254)] | Param. #1: <Network Data> |
| 22 | 2c | - | Bad parameter: other than empty payload. |

| Example | |
|---|---|
| Host → KiNOS<br>`00 : 00 : 11 : 2c : 3d`<br>Read the network data. | KiNOS → Host<br>`00 : 06 : 21 : 2c : 0e : 08 : 04 : 0b : 02 : 00 : 00`<br>Network data contains a Commissioning Data TLV with a Commissioner Session ID value of 0x0000. |

# Statistics

Show a statistics summary of MAC layer received and transmitted frames status and general device errors from the last `ifup` event.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 2d | - | No parameters. |
| **Responses** | | | |
| 21 | 2d | HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) · HEXN(4) | Param. #1: <In unknown protocols><br>Param. #2: <In errors ><br>Param. #3: <Out errors ><br>Param. #4: <In unicast frames><br>Param. #5: <In broadcast frames><br>Param. #6: <In discards ><br>Param. #7: <Out unicast frames><br>Param. #8: <Out broadcast frames><br>Param. #9: <Out discards ><br>Param. #10: <DSA errors><br>Param. #11: <Queue errors><br>Param. #12: <USB errors><br>Param. #13: <Ethernet errors><br>Param. #14: <UART errors> |
| 22 | 2d | - | Bad parameter: other than empty payload. |

## Example

Host → KiNOS

```
00 : 00 : 11 : 2d : 3c
```

KiNOS → Host

```
00 : 38 : 21 : 2d : 3a : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 02 : 00 : 00 : 00 :
17 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
01 : 00 : 00 : 00 : 1a : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
```

| | |
|---|---|
| Read the device's statistics. | `00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : 00`<br>Statistics:<br>- 2 incoming unicast frames.<br>- 23 incoming broadcast frames.<br>- 1 outgoing unicast frame.<br>- 26 outgoing broadcast frames. |

# Children Table

Show a list of device's children and information about connectivity with them.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 2e | - | No parameters. |
| **Responses** | | | |
| 21 | 2e | [LIST(HEXN(2) · DEC(4) · DEC(4) · HEXN(1) · DEC(1) · DEC(1) · ADDR(8))] | Param. #1: <Short MAC address><br>Param. #2: <Timeout><br>Param. #3: <Age><br>Param. #4: <Mode><br>Param. #5: <Link margin><br>Param. #6: <Network data version><br>Param. #7: <Extended MAC address> |
| 22 | 2e | - | Bad parameter: other than empty payload. |

## Example

Host → KiNOS

```
00 : 00 : 11 : 2e : 3f
```

Read the device's children table.

KiNOS → Host

```
00 : 2a : 21 : 2e : a6 : d4 : 01 : 00 :
00 : 00 : f0 : 00 : 00 : 00 : 8f : 0d :
3b : 4f : 2a : 3d : 54 : 97 : 2c : ff :
3a : 65 : d4 : 02 : 00 : 00 : 00 : f0 :
00 : 00 : 00 : 7f : 0f : 41 : 4f : 8a :
8f : c6 : 3b : 72 : 17 : 8e : 43
```

Device's children table consists of two children:

```
+-----------------------------------------------------------------+
| Rloc16 | Tout |  Age |  LM | Ver | Mode |     Extended Address    |
+--------+------+------+-----+-----+------+-------------------------+
| 0xd401 |  240 |  143 |  59 |  79 |  R N | 2a-3d-54-97-2c-ff-3a-65 |
| 0xd402 |  240 |  127 |  65 |  79 |  RDN | 8a-8f-c6-3b-72-17-8e-43 |
+-----------------------------------------------------------------+
```

# Socket Send

Use a previously opened socket (identified by the source port) to send arbitrary UDP data to a remote endpoint (identified by the destination address and destination port).

The maximum UDP content length is 1232 bytes, which comes from the MTU of IPv6 in Thread (1280 bytes) once the IPv6 header (40 bytes) and UDP header (8 bytes) have been excluded. Fragmentation/reassembly is not used.

**Execute**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 2f | DEC(2) · DEC(2) · ADDR(16) · HEX(1232) | Param. #1: <Source port><br>Param. #2: <Destination port><br>Param. #3: <Destination address><br>Param. #4: <UDP payload> |
| **Responses** | | | |
| 20 | 2f | - | OK. |
| 22 | 2f | - | Bad parameter. |
| 24 | 2f | - | Operation not allowed. |

**Execution conditions**

Status must be `joined`. Source port must be previously associated to a socket.

**Example**

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 1a : 10 : 2f : 8d : 30 : 39 : 1f : 90 : fd : 00 : 0d : b8 : 00 : 00 : 00 : 00 : 00 : 00 : 00 : ff : fe : 00 : 04 : 00 : 48 : 65 : 6c : 6c : 6f : 21` | `00 : 00 : 20 : 2f : 0f` |
| Send a UDP message with source port 12345, destination port 8080, destination address fd00:db8::ff:fe00:400 and payload “Hello!” (ASCII). | Command executed correctly. |

# Firmware Update

Used for sending a new firmware or patch file to the device.

A Kirale provided DFU file must be sent, excluding the DFU suffix, in successive blocks of 64 bytes in a “stop and wait ARQ” mechanism. The last block may be of smaller size. Retransmissions of not correctly acknowledged blocks are allowed. After last block has been acknowledged a reset command should be issued for the new firmware to be installed.

Up to 10 seconds of response delay might be needed for this command. The first block is numbered as 0.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 30 | DEC(2) · HEX(64) | Param. #1: <Block ID><br>Param. #2: <Firmware block> |
| Responses | | | |
| 21 | 30 | DEC(2) | Param. #1: <Block ID> |
| 22 | 30 | - | Bad parameter. |
| 24 | 30 | - | Operation not allowed. |
| 27 | 30 | - | Firmware update error. |

| Execution conditions |
|---|
| Always allowed, but status recommended to be none. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| 00 : 42 : 10 : 30 : 14 : 0e : f6 : de :<br>0e : 6a : dd : 7b : 00 : 4a : 21 : 89 :<br>25 : 6e : 79 : 47 : 59 : 54 : 8d : 84 :<br>28 : e7 : b4 : bf : cb : eb : c6 : e5 :<br>c4 : c5 : 0b : 89 : 7b : 7e : 7d : aa :<br>9c : 18 : a4 : fd : 06 : aa : b5 : 2d :<br>55 : 98 : cd : 15 : 4c : 5b : 06 : c0 : | 00 : 02 : 21 : 30 : eb : 0e : f6 |

| | |
|---|---|
| `78 : fa : a7 : ff : 98 : 09 : fe : 07 : a1 : 85 : c3 : 10 : 01 : bb : 45` | |
| Send the firmware block number 3830. | Firmware block number 3830 correctly received. |

# Hardware Mode

Control the enabled communication peripherals (Serial, USB and Ethernet).

| Value | Hardware mode description |
|---|---|
| 1 | USB and Serial peripherals enabled ( |
| 2 | USB peripheral enabled |
| 3 | Serial peripheral enabled |
| 4 | USB and Ethernet peripherals enabled |

Hardware mode changes will only take effect after the next reset.

**Important**: be aware that setting a mode which disables the Serial peripheral will make impossible to access the device any more, unless USB access is available.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 1 | Immediate | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 31 | - | No parameters. |
| **Responses** | | | |
| 21 | 31 | ENU | Param. #1: <Hardware mode> |
| 22 | 31 | - | Bad parameter. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 31 : 20
```

Read the hardware mode.

KiNOS → Host

```
00 : 01 : 21 : 31 : 10 : 01
```

Hardware mode: 1 (USB and Serial peripherals enabled).

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 31 | ENU | Param. #1: <Hardware mode> |
| | | Responses | |
| 20 | 31 | - | OK. |
| 22 | 31 | - | Bad parameter. |
| 24 | 31 | - | Operation not allowed. |

### Modification conditions

Always allowed.

### Example

Host → KiNOS

```
00 : 01 : 10 : 31 : 23 : 03
```

Set the hardware mode to 3 (Serial peripheral enabled).

KiNOS → Host

```
00 : 00 : 20 : 31 : 11
```

Hardware mode correctly stored.

# LED Mode

Enable or disable the status LED.

LED mode changes will only take effect after the next reset.

When enabled, the LED indicates different states of the device:

| Blinking type | Description |
|---|---|
| Slow | Not connected to a network |
| Normal | Connecting to a network |
| Fast | Receiving firmware update blocks |
| Super fast | In bootloader – during firmware update applying |
| Steady | Connected to a network |

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 1 (on) | Immediate | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 32 | - | No parameters. |
| **Responses** | | | |
| 21 | 32 | ENU | Param. #1: <LED mode>, 0 = 'off', 1 = 'on'. |
| 22 | 32 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 32 : 23
```

Read the LED mode.

KiNOS → Host

```
00 : 01 : 21 : 32 : 13 : 01
```

LED mode value: 1 ('on').

**Write (on)**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 32 | - | Write with no parameters means set to 'on'. |
| **Responses** | | | |
| 20 | 32 | - | OK. |

| | | | |
|---|---|---|---|
| 22 | 32 | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 00 : 10 : 32 : 22` | `00 : 00 : 20 : 32 : 12` |
| Set the LED mode to ‘on’. | LED mode successfully stored. |

## Delete (off)

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 32 | - | Delete with no parameters means set to ‘off’. |
| **Responses** | | | |
| 20 | 32 | - | OK. |
| 22 | 32 | - | Bad parameter: other than empty payload. |

### Modification conditions

Always allowed.

### Example

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 00 : 12 : 32 : 20` | `00 : 00 : 20 : 32 : 12` |
| Set the LED mode to ‘off’. | LED mode successfully stored. |

# Vendor Name

Vendor name, used in Thread Commissioning.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| Kirale Technologies | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 33 | - | No parameters. |
| **Responses** | | | |
| 21 | 33 | STR(2,33) | Param. #1: <Vendor name> |
| 22 | 33 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 33 : 22
```

Show vendor name.

KiNOS → Host

```
00 : 14 : 21 : 33 : 3e : 4b : 69 : 72 :
61 : 6c : 65 : 20 : 54 : 65 : 63 : 68 :
6e : 6f : 6c : 6f : 67 : 69 : 65 : 73 :
00
```

Vendor name: ‘Kirale Technologies’.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 33 | STR(1,32) | Param. #1: <Vendor name> |
| **Responses** | | | |
| 20 | 33 | - | OK. |
| 22 | 33 | - | Bad parameter. |

### Modification conditions

Always allowed.

Example

| Host → KiNOS | KiNOS → Host |
| --- | --- |
| `00 : 04 : 10 : 33 : 2d : 41 : 43 : 4d : 45` | `00 : 00 : 20 : 33 : 13` |
| Set the vendor name to ‘ACME’. | Vendor name stored. |

# Vendor Model

Hardware version of the vendor application, used in Thread Commissioning.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| KTWM102 Module | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 34 | - | No parameters. |
| **Responses** | | | |
| 21 | 34 | STR(2,33) | Param. #1: <Vendor model> + EOS |
| 22 | 34 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 34 : 25
```

Show vendor model.

KiNOS → Host

```
00 : 0f : 21 : 34 : 36 : 4b : 54 : 57 :
4d : 31 : 30 : 32 : 20 : 4d : 6f : 64 :
75 : 6c : 65 : 00
```

Vendor model: ‘KTWM102 Module’.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 34 | STR(1,32) | Param. #1: <Vendor model> |
| **Responses** | | | |
| 20 | 34 | - | OK. |
| 22 | 34 | - | Bad parameter. |

### Modification conditions

Always allowed.

Example

Host → KiNOS

```
00 : 08 : 10 : 34 : 77 : 4d : 79 : 20 :
4d : 6f : 64 : 65 : 6c
```

Set the vendor model to ‘My Model’.

KiNOS → Host

```
00 : 00 : 20 : 34 : 14
```

Vendor model stored.

# Vendor Data

Vendor application information data, used in Thread Commissioning.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | Immediate | No |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 11 | 35 | - | No parameters. |
| | | Responses | |
| 21 | 35 | [STR(2,65)] | Param. #1: <Vendor data> + EOS |
| 22 | 35 | - | Bad parameter: other than empty payload. |

Example

Host → KiNOS

```
00 : 00 : 11 : 35 : 24
```

Show vendor data.

KiNOS → Host

```
00 : 00 : 21 : 35 : 14
```

Vendor data is empty.

**Write**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| | | Command | |
| 10 | 35 | STR(1,64) | Param. #1: <Vendor data> |
| | | Responses | |
| 20 | 35 | - | OK. |
| 22 | 35 | - | Bad parameter. |

Modification conditions

Always allowed.

Example

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 0a : 10 : 35 : 69 : 61 : 73 : 64 : 66 : 67 : 31 : 32 : 33 : 34 : 35`<br>Set the vendor data to ‘asdfg12345’. | `00 : 00 : 20 : 35 : 15`<br>Vendor data stored. |

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 35 | - | No parameters. |
| **Responses** | | | |
| 20 | 35 | - | OK. |
| 22 | 35 | - | Bad parameter. |

### Modification conditions

Always allowed.

### Example

| Host → KiNOS | KiNOS → Host |
|---|---|
| `00 : 00 : 12 : 35 : 27`<br>Clear the vendor data. | `00 : 00 : 20 : 35 : 15`<br>Vendor data cleared. |

# Vendor Software Version

Software version of the vendor application, used in Thread Commissioning.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| KiNOS v1.2 | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 36 | - | No parameters. |
| **Responses** | | | |
| 21 | 36 | STR(2,17) | Param. #1: <Vendor software version> + EOS |
| 22 | 36 | - | Bad parameter: other than empty payload. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 36 : 27
```

Show vendor software version.

KiNOS → Host

```
00 : 0b : 21 : 36 : 17 : 4b : 69 : 4e :
4f : 53 : 20 : 76 : 31 : 2e : 32 : 00
```

Vendor software version: ‘KiNOS v1.2’

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 36 | STR(1,16) | Param. #1: <Vendor software version> |
| **Responses** | | | |
| 20 | 36 | - | OK. |
| 22 | 36 | - | Bad parameter. |

### Modification conditions

Always allowed.

Example

Host → KiNOS

```
00 : 0a : 10 : 36 : 20 : 4d : 79 : 41 :
70 : 70 : 20 : 76 : 31 : 2e : 30
```

Set the vendor software version to ‘MyApp v1.0’.

KiNOS → Host

```
00 : 00 : 20 : 36 : 16
```

Vendor software version stored.

# Active Timestamp

Timestamp for the active network configuration.

| 0-47 | 48-62 | 63 |
|---|---|---|
| UNIX time value | Ticks | U bit |

UNIX time value: Unsigned integer that encodes seconds.

Ticks: Unsigned integer that encodes UNIX time value in ticks of 32.768 kHz.

U bit: Indicates the time was obtained from an authoritative source: either NTP (Network Time Protocol), GPS (Global Positioning System), cell network, or other method.

The active timestamp can be acquired from the network as well.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| 0x0000000000010000 | When `joined` | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 37 | - | No parameters. |
| **Responses** | | | |
| 21 | 37 | HEXN(8) | Param. #1: <Active timestamp> |
| 22 | 37 | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 37 : 26
```

Show active timestamp.

KiNOS → Host

```
00 : 08 : 21 : 37 : 1f : 00 : 00 : 00 :
00 : 00 : 01 : 00 : 00
```

Active timestamp: 0x0000000000010000 (1 second).

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 37 | HEXN(8) | Param. #1: <Active timestamp> |
| **Responses** | | | |
| 20 | 37 | - | OK. |
| 22 | 37 | - | Bad parameter. |

### Modification conditions

Status must be none, except `none -saved configuration`.

### Example

Host → KiNOS

```
00 : 08 : 10 : 37 : 3b : 00 : 00 : 00 :
00 : 00 : 14 : 00 : 00
```

Set the active timestamp to 20 seconds.

KiNOS → Host

```
00 : 00 : 20 : 37 : 17
```

Active timestamp saved.

# Named Ping

Send an ICMPv6 echo request to a specified destination domain with a specified payload length (random payload).

The device will automatically try to resolve the domain name to an address using DNS, or lookup in its DNS cache.

The ICMPv6 echo reply, if received, will be delivered to the host as an asynchronous notification. Depending on the role and the type of destination address, there could be a destination unreachable notification.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 38 | STRN(32) · DEC(2) | Param #1: <Destination domain><br>Param #2: <Payload size>, 0 to 1232. |
| Responses | | | |
| 20 | 38 | - | OK. |
| 22 | 38 | - | Bad parameter. |

**Execution conditions**

Status must be `joined`.

**Example**

Host → KiNOS

```
00 : 22 : 10 : 38 : 49 : 6b : 69 : 72 :
61 : 6c : 65 : 2e : 63 : 6f : 6d : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :
00 : 00 : 00 : 00 : 00 : 00 : 14
```

Send an ICMPv6 echo request with 20 bytes payload size to the destination domain ‘kirale.com’.

KiNOS → Host

```
00 : 00 : 24 : 38 : 1c
```

Command not allowed.

# Named Socket Send

Use a previously opened socket (identified by the source port) to send arbitrary UDP data to a remote endpoint (identified by the destination domain and destination port).

The device will automatically try to resolve the domain name to an address using DNS, or lookup in its DNS cache.

The maximum UDP content length is 1232 bytes, which comes from the MTU of IPv6 in Thread (1280 bytes) once the IPv6 header (40 bytes) and UDP header (8 bytes) have been excluded. Fragmentation/reassembly is not used.

| Execute | | | |
|---|---|---|---|
| TYPE | CMD | PAYLOAD | COMMENTS |
| Command | | | |
| 10 | 39 | DEC(2) · DEC(2) · STRN(32) · HEX(1232) | Param. #1: <Source port><br>Param. #2: <Destination port><br>Param. #3: <Destination domain><br>Param. #4: <UDP payload> |
| Responses | | | |
| 20 | 39 | - | OK. |
| 22 | 39 | - | Bad parameter. |
| 24 | 39 | - | Operation not allowed. |

| Execution conditions |
|---|
| Status must be `joined`. Source port must be previously associated to a socket. |

| Example | |
|---|---|
| Host → KiNOS | KiNOS → Host |
| `00 : 27 : 10 : 39 : 0d : ff : 90 : 26 :`<br>`94 : 6b : 69 : 72 : 61 : 6c : 65 : 2e :`<br>`63 : 6f : 6d : 00 : 00 : 00 : 00 : 00 :`<br>`00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :`<br>`00 : 00 : 00 : 00 : 00 : 00 : 00 : 00 :`<br>`00 : ab : cd : ef` | `00 : 00 : 20 : 39 : 19` |

| | |
|---|---|
| Send a UDP message with source port 65424, destination port 9876, destination domain ‘kirale.com’ and payload 0xabcdef. | Command executed correctly. |

# Services Status

Show the status (on/off) of different network services: DHCP, DNS and NTP.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 3a | - | No parameters. |
| **Responses** | | | |
| 21 | 3a | ENU · ENU · ENU | Param. #1: <DHCP status>, 0 = 'off, 1 = 'on'.<br>Param. #2: <DNS status>, 0 = 'off, 1 = 'on'.<br>Param. #3: <NTP status>, 0 = 'off, 1 = 'on'. |
| 22 | 3a | - | Bad parameter: other than empty payload. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 3a : 2b
```

Show the services status.

KiNOS → Host

```
00 : 03 : 21 : 3a : 18 : 00 : 00 : 00
```

DHCP is off, DNS is off, NTP is off.

# Provisioning URL

Configuration of the Joiner's Provisioning URL used in Thread Commissioning.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | Immediate | No |

## Read

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 3b | - | No parameters. |
| **Responses** | | | |
| 21 | 3b | [STR(2,65)] | Param. #1: <Provisioning URL> |
| 22 | 3b | - | Bad parameter. |

### Example

Host → KiNOS

```
00 : 00 : 11 : 3b : 2a
```

Read the provisioning URL.

KiNOS → Host

```
00 : 00 : 21 : 3b : 1a
```

Provisioning URL is empty.

## Write

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 10 | 3b | STR(1,64) | Param. #1: <Provisioning URL> |
| **Responses** | | | |
| 20 | 3b | - | OK. |
| 22 | 3b | - | Bad parameter. |
| 24 | 3b | - | Operation not allowed. |

### Modification conditions

Status must be none.

### Example

Host → KiNOS

```
00 : 0a : 10 : 3b : 76 : 6b : 69 : 72 :
61 : 6c : 65 : 2e : 63 : 6f : 6d
```

Set the provisioning URL to ‘kirale.com’.

KiNOS → Host

```
00 : 00 : 20 : 3b : 1b
```

Provisioning URL successfully stored.

## Delete

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 12 | 3b | - | No parameters. |
| **Responses** | | | |
| 20 | 3b | - | OK. |
| 22 | 3b | - | Bad parameter. |
| 24 | 3b | - | Operation not allowed. |

### Modification conditions

Status must be none.

### Example

Host → KiNOS

```
00 : 00 : 12 : 3b : 29
```

Delete the provisioning URL

KiNOS → Host

```
00 : 00 : 24 : 3b : 1f
```

Command not allowed.

# Commissioner Session ID

Show the active commissioner session ID.

| Default value | NVM storage | Restored by `clear` |
|---|---|---|
| None | No | Yes |

**Read**

| TYPE | CMD | PAYLOAD | COMMENTS |
|---|---|---|---|
| **Command** | | | |
| 11 | 3c | - | No parameters. |
| **Responses** | | | |
| 21 | 3c | [HEXN(2)] | Param. #1: <Commissioner session ID> |
| 22 | 3c | - | Bad parameter. |

**Example**

Host → KiNOS

```
00 : 00 : 11 : 3c : 2d
```

Read the commissioner session ID.

KiNOS → Host

```
00 : 02 : 21 : 3c : 1e : 00 : 01
```

Commissioner session ID: 0x0001.

*END OF THIS DOCUMENT*