Common NUR module / API error codes

Click the link on right for details.

Code	Name	Is
1	INVALID_COMMAND	Command not recognized.
2	INVALID_LENGTH	Invalid command/parameter length.
3	PARAMETER_OUT_OF_RANGE	Parameter range error.
5	INVALID_PARAMETER	One or more parameter errors.
13	NOT_READY	Not ready to execute command or execution interrupted.
16	G2_GENERAL_ERROR	Gen2 version 2 sub-command error.
32	ERROR_NO_TAG	No tags or tag found.
34	G2_ERROR_SELECT	Tag singulation error.
36	G2_ERROR_ACCESS	Tag accessing error.
48	G2_ERROR_READ	Generic read error.
64	G2_ERROR_WRITE	Generic write error.
65	G2_ERROR_WR_PART	Write succeeded partially.
66	G2_TAG_ERROR_RESP	Tag backscattered an error.
Several	Tags' error replies	Tag error interpreted by the API.

INVALID_COMMAND (0x01)

The error is trivial; the module does not recognize the command. More specifically:

- FW version mismatch; a command requested is not implemented in this version of the FW
- Also the current module configuration may not support the command whereas some other configuration may support it
- Error in the host application, check the used command's value

INVALID_LENGTH (0x02)

Reason for this error is that the command's parameter length is not what is expected. Examples where this error may occur:

- API / FW mismatch: e.g. the used module setup structure is not what the FW expects due to newer implementation in the API
- Read, write, lock or kill parameters do not match what was expected (genrally applies to any Gen2 command)
- On the API side the parsing of the antenna mapping fails

PARAMETER_OUT_OF_RANGE (0x03)

On the module's side this error currently occurs (in FW version 5.5-A and higher) when the Gen2 v2 authenticate command is trying to receive longer response than 1023 bits (HW restriction, no workaround other than implementing a shorter reply scheme for the request).

On API side this error occurs when

- When number of channels is 0 (other parameters are parsed OK) after parsing a custom hoptable from an INI-file
- A tag data entry is being retrieved with an index higher than number of tags currently stored
- When getting or deleting a WLAN profile with index higher than 7

INVALID_PARAMETER (0x05)

The reason is trivial; either the API or module FW has interpreted the given parameter to any command as invalid. This may be triggered due to the length of a parameter and/or the value of the parameter.

NOT_READY (0x0D, 13)

Generally this error is received from the module when storing or updating some internal information fails. This can happen due to incorrect parameters as well; the module is stating that this specific configuration cannot be saved. Examples:

- Ethernet clock parameter length error or getting/setting its value failed
- Getting or setting the Ethernet configuration
- EPC enumeration was tried to run without stopping other continuous commands first
- EPC enumeration is running when other continuous command is tried to be started
- Scratch data save fails
- Any of the tuning commands fails

Examples on the API side:

- Physical antenna enabling or antenna ID mapping fails
- Getting physical antenna map fails (currently not stored by API)
- Physical antenna enable query fails
- Unexpected termination while communicating with the module (transport error as well)

G2_GENERAL_ERROR (0x10, 16)

On the module's side error occurs when Gen2 v2 sub-command fails internally. This error should never show up via an API call, the error may occur with an embedded implementation.

ERROR_NO_TAG (0x20, 32)

This error occurs when

- A single inventory returns no tags
- A scan single command either cannot find a tag or there are too many tags in the field (one expected)

G2_ERROR_SELECT (0x22, 34)

Tag singulation for any Gen2 operation failed. The operation can be anything that currently requires tag addressing such as read, write, lock kill and so on. Reasons for this error typically are

- Given singulation data is invalid; no tag replies to it i.e. do not recognize it to be theirs
- Low RF power or tag too far, erroneous reply received when singulating the tag
- Too many tags in the field when trying to perform an operation such read, write etc. without any singulation parameters
- The given singulation parameters are correct but the tag was removed from the field
- Also if the given parameters are correct it is possible that the selection criteria matches to more than one tag; the reader will get erroneous replies thus cannot select the tag

G2_ERROR_ACCESS (0x24, 36)

This error occurs when the given operation requires the use of an access password in order to perform the following operation that can be for example read or write. Error typically occurs when the given password is incorrect.

G2_ERROR_READ (0x30, 48)

This is general, unspecified read error. Typical reason for the error is that the read reply was corrupted or not received at all.

G2_ERROR_WRITE (0x40, 64)

This is general, unspecified write error. Typical reason for the error is that the write reply was corrupted or not received at all. An RF interference or tag moving out in the middle of the reply can cause this error. When the write reply cannot be interpreted by the reader, it then tries to verify the last write result by reading the data back comparing it to the data expected. If this read back fails as well, then the error falls to this one, an error that cannot be interpreted.

G2_ERROR_WR_PART (0x41, 65)

This error occurs when the write succeeded partially i.e. some of the given data was successfully written but at some point during the write cycles an error occurred. This may be due that some the data would've been written to a memory location that does not exist. Also if during the write cycles an RF interference would cause similar situation as with G2_ERROR_WRITE then, if some of the data was written OK, this error occurs.

G2_TAG_ERROR_RESP (0x42, 66)

The tag replied with an error. These errors are specified in the Gen2 specification and they include such things as memory does not exist, power too low or that the memory is locked for reading or writing.

With custom/proprietary commands the tag manufactures ca also specify their own error codes.

Tag errors interpreted by the API

The NUR API maps the following tag errors to codes shown below:

Reply error	API error	Meaning
3	0x100E, 4110	Memory overrun; memory does not exist
4	0x100F, 4111	Memory locked, not accessible with current parameters
0x0B (11)	0x1010, 4112	Insufficient power to execute the command
0x0F (11)	0x1011, 4113	Non-specific error (specific error codes not supported)
Others	0x42, 66	NUR_ERROR_G2_TAG_RESP (not translated)