User Guide

EtherNet/IP Interface Communication Protocol Via Direct Connection

Supported technologies

Cx-Series

D/F-Series

Gx-Series

Contents

0 Version History

Revision	Date	Changes
R01	Nov-2024	First release.

THIS PAGE INTENTIONALLY LEFT BLANK

1 EtherNet/IP Command Reference

1.1 Introduction

EtherNet/IP is one of the leading industrial protocols and is widely used. EtherNet/IP is an industrial network protocol that provides direct communication from PLCs (Programmable Logic Controller) to Domino coders supporting that standard. Rockwell PLCs build a specialty here. Due to the partnership between Rockwell and Domino, special support (e.g. a profile in Studio 5000) is available.

EtherNet/IP allows for full control over Domino coding and marking equipment, and its messaging, through this standard industrial protocol. The printer/coder can also relay performance data, allowing for predictability and pre-emptive maintenance using implicit and explicit messaging.

With some Domino products, two implementations of EtherNet/IP are available. Some coder series have the capability to directly integrate an EtherNet/IP port in the controllers (e.g. D/F-Series). Alternatively, an integration via an external interface can be offered. The external interface offers a broader support and is specifically optimised for ease of use on Rockwell PLCs compatible to the Studio 5000 software. To mount the external interface and run the EtherNet/IP software an Interface Controller from Domino is required.

The two EtherNet/IP implementations are not identical. **This document is the user guide for EtherNet/IP using an interface directly implemented in the coding system controller**.

1.2 Hardware Requirements for EtherNet/IP via Interface Directly in the Coding System Controller

This document describes the use of EtherNet/IP on Domino coding systems via an interface directly placed in the controller of the coding system. Additional hardware to mount the EtherNet/IP adapter (e.g Automation Gateway 100) is not used. This is only possible if Domino Coders are used that can host an EtherNet/IP interface directly. This document only deals with the implementation of EtherNet/IP in such systems.

For information about an EtherNet/IP connection in Domino print system using a separate Interface Controller (e.g Automation Gateway), refer to: Doc-0023889 - User Guide: EtherNet/IP Interface Communication Protocol Using a Domino Interface Controller

1.3 General

EtherNet/IP is an industrial network protocol that adapts the Common Industrial Protocol (CIP) to standard Ethernet. For that reason the CIP specifications are valid. EtherNet/IP follows CIP command, message, and error structure.

EtherNet/IP is an application layer protocol that is transferred inside TCP/IP and UDP/IP packets.

Domino is implementing its vendor specific (explicit) commands based on its Dynamark protocol. Due to the specific EtherNet/IP protocol requirements this shall be explained in more detail.

1.4 Rockwell PLC Used in this Documentation

This documentation is based on using a Rockwell PLC. This leads to some special items:

- The partnership Domino established with Rockwell enables using implicit commands without loading an EDS file upfront. The required configuration is already available within the Studio5000 software as a printer specific profile. Nevertheless, the concept of activating implicit messages is explained in 1.9 Implicit Messages.
- The EDS provided by Domino supports Rockwell CompactLogix and Rockwell ControlLogix PLCs.
- There are some special shortcuts and settings used in Rockwell PLCs which may not be available in other PLCs. E.g. the "$" symbol is used preceding a byte (hex) value in tags. And there are special bytes that have some shortcut like "$t" instead of "$09" which represents the value "9". In instances where this document includes examples showing hex code, Rockwell shortcuts will be utilized.
- In Studio 5000 the representation of a data type on the screen can be changed e.g. by setting the "Style" of an SINT to ASCII. This makes sense in case a character is encoded in a SINT or a succession of SINT represents a string. That way the SINT values will show as their translation to ASCII on the screen. With the examples used below and in the screen copies in this document we have done this for readability. But the original unaltered entry is also given.

1.5 Little Endian

All numeric values are using little endian. This is in consensus with CIP convention also followed by Rockwell.

That means that the least significant bytes are given first. This is especially true for the length parameter. In the above example of a 12-byte length for a string the value is "$0b$00$00$00" as "$0b" is the hex representation of "12" and "$00$00$00" is the required to fill the four-byte length of DINT.

Another example may be the decimal number "1000" into a DINT. The hex conversion in little endian is "$E8$03$00$00".

Because we use little endian everywhere this formatting will not be mentioned below.

1.6 Lengths and UTF8 Encoded Strings and Numbers

Strings provided as parameters to a command are UTF8 encoded if not explicitly mentioned. A length value is preceding the UTF8 encoded string. This length defines how many bytes follow containing the string. In this the length is usually given as DINT little endian. The string is then encoded as row of SINT (Array of SINT) containing the characters encoded in UTF8. As example the string 'a parameter' shall be encrypted using the described method. The result will be:
$0b$00$00$00$61$20$70$61$72$61$6d$65$74$65$72.

Here "$0b$00$00$00" is the length of the string as DINT little endian. "$61$20$70$61$72$61$6d$65$74$65$72" bytes code the string "a parameter" as an Array of SINT . There is no discriminator used in between.

In most PLC programs it is possible to display UTF8 encoded text as ASCII characters. That way it is possible to show 'a parameter' human readable in a tag. For ease of reading this has been done in the following description using the Rockwell Studio 5000 software. The line above will then show as $0b$00$00$00a parameter in the tag.

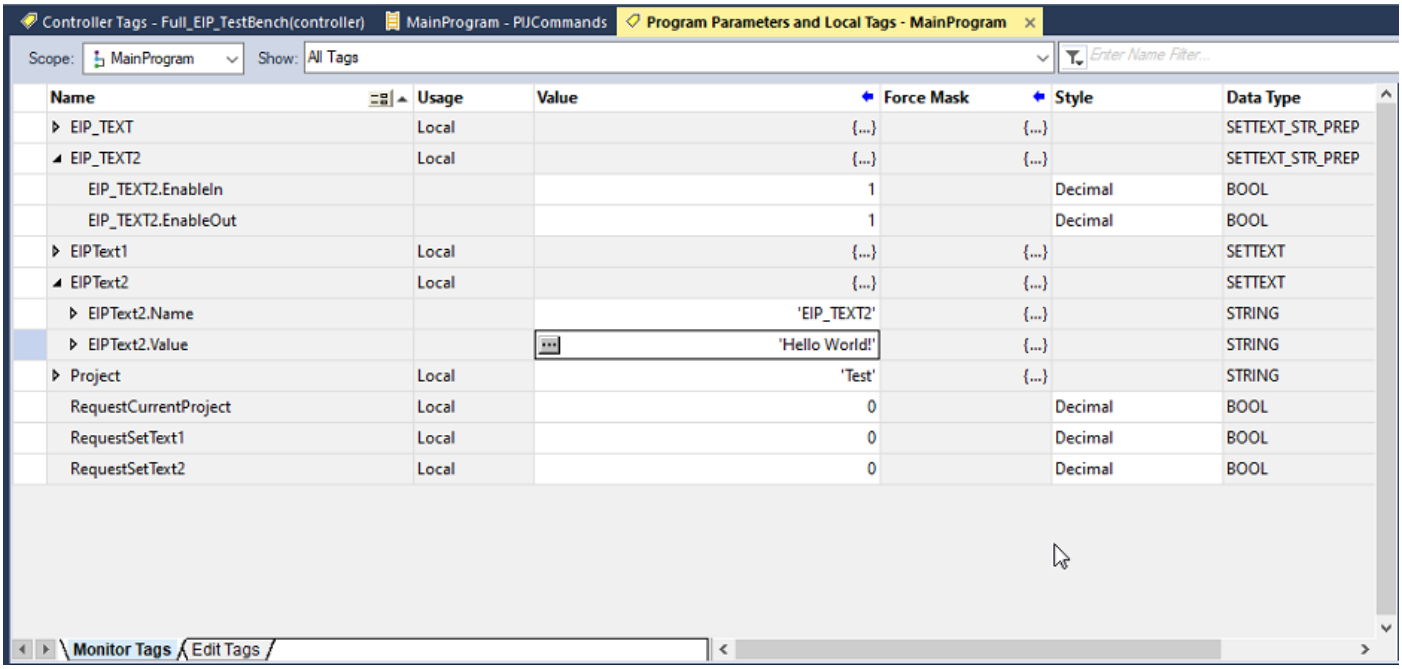
It is important to recognise that also some numbers in parameter are encoded this way. An example is the command SETENCODERDELAY. After a global parameter ("0" or "1") the length of the value to be set as DINT must be given. This is then followed by a Array of SINT encoding the number characters in UTF8. In numbers with decimal points the "." must be used. The unit mm must not be part of the parameter as the command always assumes it to be mm. Therefore, to set the encoder delay to 2.05 mm the lengths is $04$00$00$00 and the value of "2.05" must be $32$2e$30$35.

Note: The encoder delay must have two decimals.

1.7 How to Fill Tags: An Example

Many commands Domino implemented in EtherNet/IP require a variable and its value as parameters. An example is the command 0x53 - SETTEXT that allows to set the value of a text string as variable.

In our example the variable name shall be: "EIP_TEXT2"



Name	Usage	Value	Force Mask	Style	Data Type
EIP_TEXT	Local	{...}	{...}		SETTEXT_STR_PREP
EIP_TEXT2	Local	{...}	{...}		SETTEXT_STR_PREP
EIP_TEXT2.EnableIn		1		Decimal	BOOL
EIP_TEXT2.EnableOut		1		Decimal	BOOL
EIPText1	Local	{...}	{...}		SETTEXT
EIPText2	Local	{...}	{...}		SETTEXT
EIPText2.Name		'EIP_TEXT2'	{...}		STRING
EIPText2.Value		'Hello World!'	{...}		STRING
Project	Local	'Test'	{...}		STRING
RequestCurrentProject	Local	0		Decimal	BOOL
RequestSetText1	Local	0		Decimal	BOOL
RequestSetText2	Local	0		Decimal	BOOL

The data that shall fill the content is "Hello World".

As can be seen in the SETTEXT command definition the length of the variable, the variable, the length of the content and the content are the parameters required (see 0x53 - SETTEXT).

The variable "EIP_TEXT2" has 9 bytes length The DINT little endian encoding of the length of the Arry of SINT is therefore t00$00$00

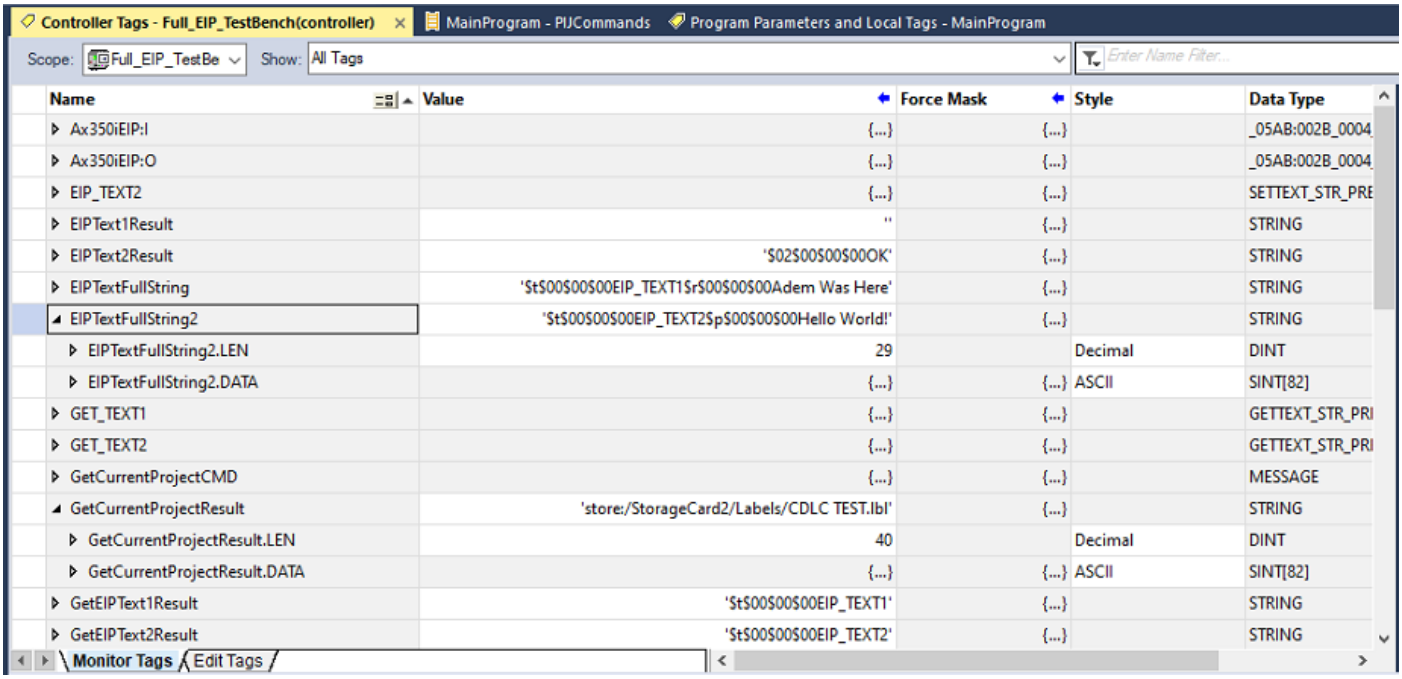
Note: "$t" are special characters that are used to encode the value 9 in Rockwell PLCs.

This content "Hello world" has 11 bytes lengths. The DINT little endian encoding of the length of the Arry of SINT is therefore p00$00$00

Note: "$p" are special characters that are used to encode the value 11 in Rockwell PLCs.

This needs to be entered into the tag **without any blanks or other separators**.

The tag must therefore contain "t00$00$00EIP_TEXT2p00$00$00Hello World". In case the style for SINT has not been changed to ASCII the full tag reads:
"t00$00$00$45$49$50$5F$54$45$58$54$32$p$00$00$00$48$65$6C$6C$6F$20$57$6F$72$6C$64".



1.8 Explicit Messages

Explicit Messaging Connections provide generic, multipurpose communication paths between two devices. Explicit Messages are exchanged across Explicit Messaging Connections. Explicit Messages are used to command the performance of a particular task and to report the results of performing the task. Explicit Messaging provides the means by which typical request/response-oriented functions are performed.

Explicit messages are used when an application requires large amounts of data. This messaging method is also a preferred choice when requiring optimising bandwidth as the data is only requested when necessary. Explicit messages are non-time critical and are request/reply in nature. Explicit messaging allows for transferring of information "on demand" for items that don't change over a short period of time and are not time sensitive. To use explicit messages programming of the PLC is required. A control logic designer will have to handle the request of data, handshaking, acknowledging the data and the business logic associated to the data in the controller.

All explicit commands defined by Domino are in Class 0x64, 0x65 and 0x66.

The selection of explicit or implicit messaging often depends on the choice of field devices, as each may support only one messaging mode.

1.9 Implicit Messages

Implicit messages are sent by the client (I/O Adapters) in regular time intervals. These intervals and properties are set during the set-up process. There is no inquiry sent by the PLC controller (client) to request any action like with explicit commands. Implicit messages are ideal for high speed and real-time applications. An advantage of using implicit messages is that the PLC controller organises the data in a table and the information is updated regularly. Implicit messaging is time critical and reserved for specific real-time and I/O data.

To work with implicit messages the PLC controller must be configured accordingly. This can be done by loading the EDS file from the Domino software package into the Rockwell PLC controller. The same can be done with other PLCs from various vendors. How the EDS file must/can be setup depends on the PLC manufacturer. Please refer to the PLC vendors documentation. The EDS file for Rockwell is part of Domino's EtherNet/IP software package but should not be required due to the availability of profiles in Studio 5000.

It is important to recognize that different profiles for the various printers of Domino exist. Accordingly, the available set of implicit messages varies with the connected printer. A full list of the messages available in the implicit profiles of printers is listed in 5 APPENDIX A: Implicit Messages

.

The time increments implicit information is updated in can be changed in the PLCs setup.

With Rockwell PLCs the set-up of implicit messages is easier. Please refer to 1.10 A Rockwell Specialty for Domino

1.10 A Rockwell Specialty for Domino

Domino is a Rockwell partner. This is a unique positioning that enables Domino to provide implicit messages for use with Domino devices by activating a Device Profile in the Studio5000 software suite. Loading the profiles configures the implicit commands in the PLC.

The Profiles can do more. Apart from loading the set-up for implicit commands, profiles can also load a set of some typical coder parameters.

The implicit messages available in the Domino device using direct adapter in the coding system controller's profiles on a Rockwell PLC are listed in 5 APPENDIX A: Implicit Messages.

2 Status (Error) Codes

As EtherNet/IP is CIP based the same error code structure applies. The error codes are separated in a general error message and a more detailed refined error code. The General Status Codes available are the CIP codes. Error code 0x1F (vendor specific error) allows Domino to define vendor-specific Extended Error Codes. Both the supported CIP General Status Codes and the Extended Error codes are given in 6 APPENDIX B: Status (Error) Messages.

3 Data Types

Type	Number of bytes
BOOL	1 (0x00 or 0x01)
SINT	1
INT	2
DINT	4
LINT	8
FLOAT	4
Array of SINT	Length defined by length parameter up front. Usually UTF8 encoded

It is important to recognize that all INT, DINT and LINT values must be given in 1.5 Little Endian. That means that the least relevant number must be given or is returned first.

4 Returning Data

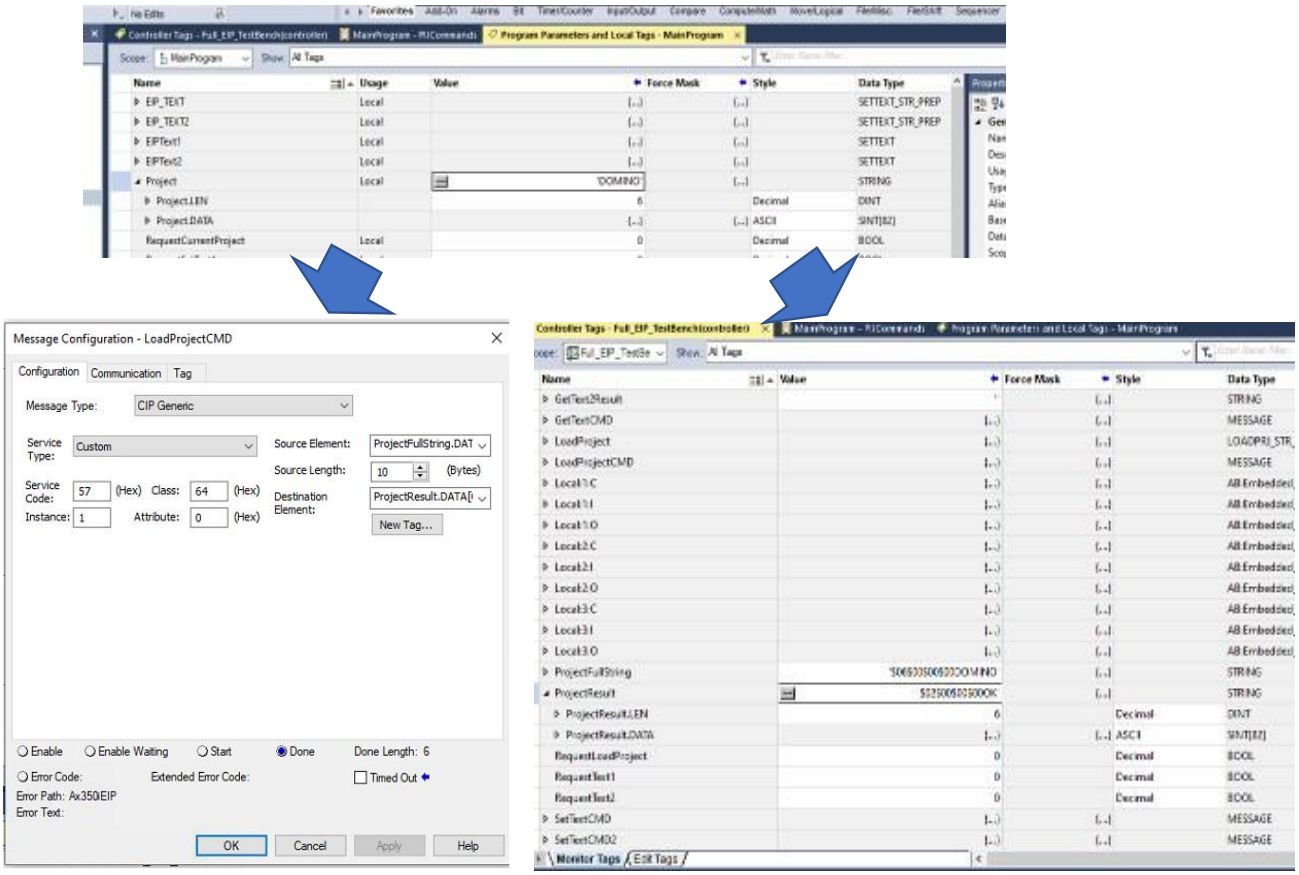
Successful explicit command requests can return different results depending on their structure. In some instances, just an "OK" is returned. This is true for all "SET" commands. Other commands may return a string. As an Array of SINT. Yet other commands may return a number or more than just one Array of SINT. To make it easier to illustrate the different result schemes we can put them into categories.

All the results are illustrated using The Studio 5000 software from Rockwell (Version 33.01.00, year 2021).

Additionally, commands can fail under certain conditions such as incorrect arguments, invalid operation in the device, or a command is not supported. The errors are reported with a number which is an INT type.

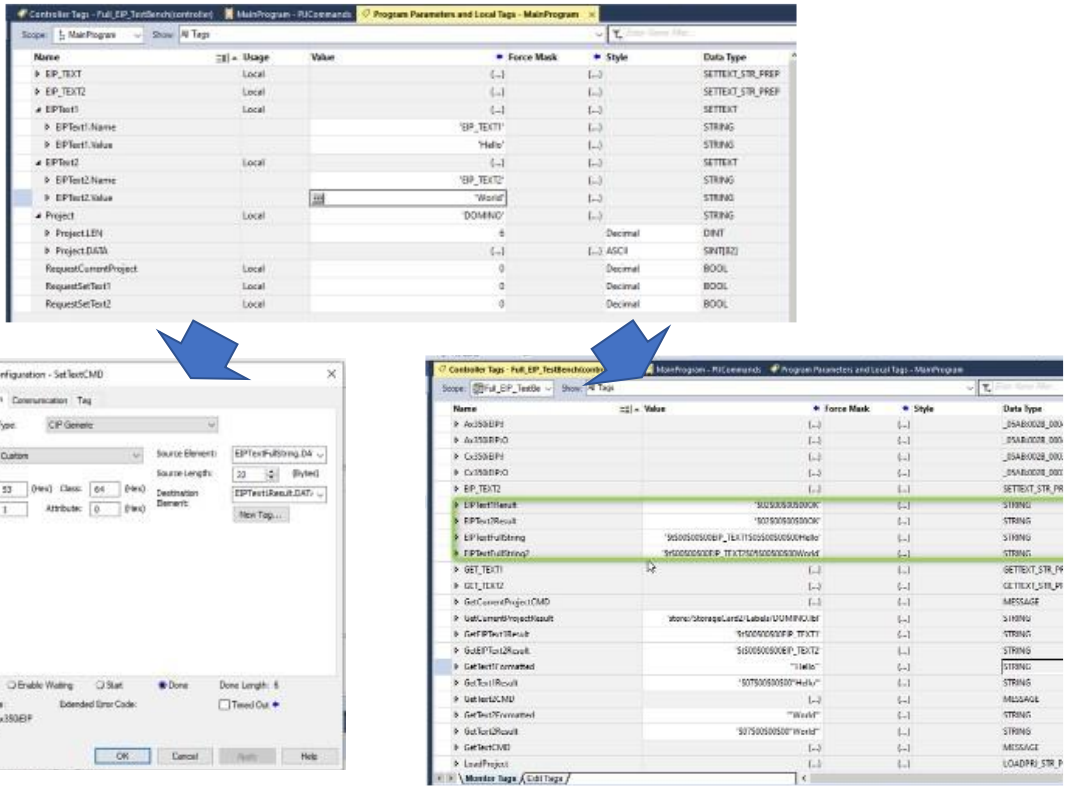
4.1 Category A: "OK" is Returned

All commands starting with a "SET" belong to this category. Another example for this category is the "Loadproject" command. It requires a string structured as Array of SINT preceded but its length specifying the project name. In the example the valid project name is "Domino". When sent to the printer this results into a "Done" flag and in parallel the result is "OK" in the Tag screen. The Array of SINT "OK" is preceded by the length of the Array given as DINT little endian.



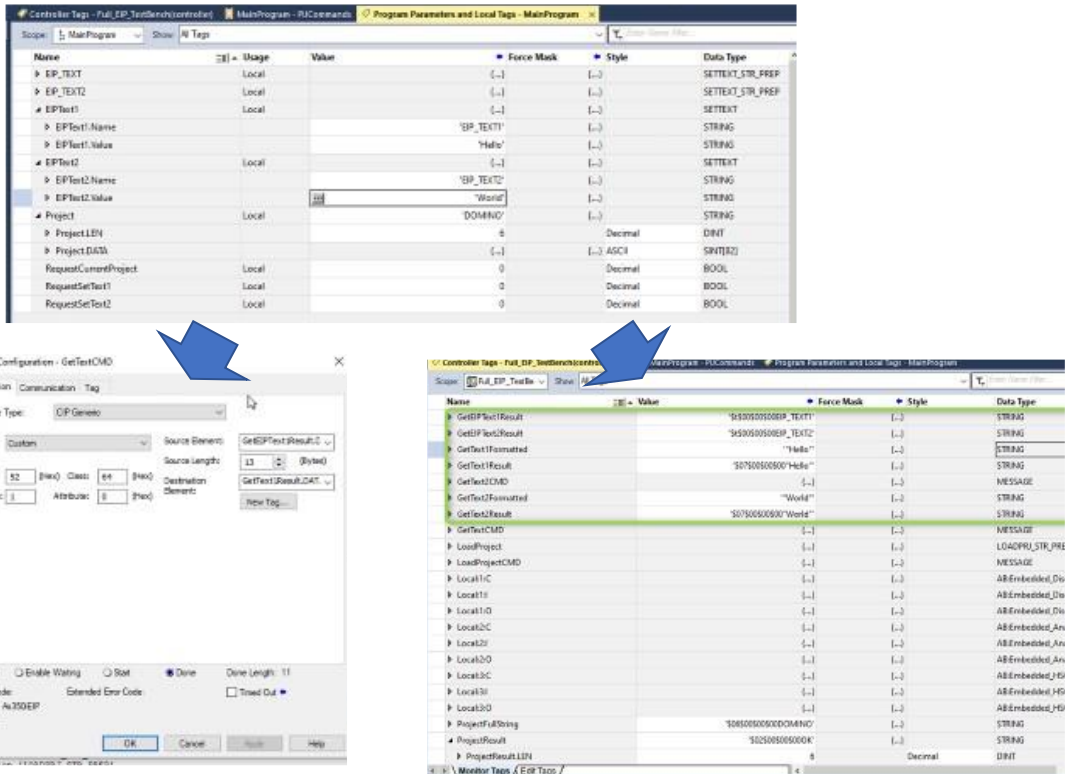
4.2 Category B: Numbers are Returned

An example may be the GETBUFFERSTATUS command. If successfully executed it returns a DINT number in little endian that describes the number of buffer entries left, this is the simplest result of this result category. In other cases, some float numbers are returned. For these details, please reference the command description.



4.3 Category C: Content is Returned as Array of SINT

Commands like GETTEXT return a string structured as Array of SINT if successfully executed. In the example below two variables EIPText1 and EIPText2 are queried. The result is a "done" flag and in parallel the content of the variable will be returned preceded by the length in DINT little endian (see 1.7 How to Fill Tags: An Example). The content itself is returned as a UTF8 encoded Array of SINT. There is no discriminator between the DINT specifying the length of the Array of SINT and the string as Array of SINT itself. This is one of the more complex results.



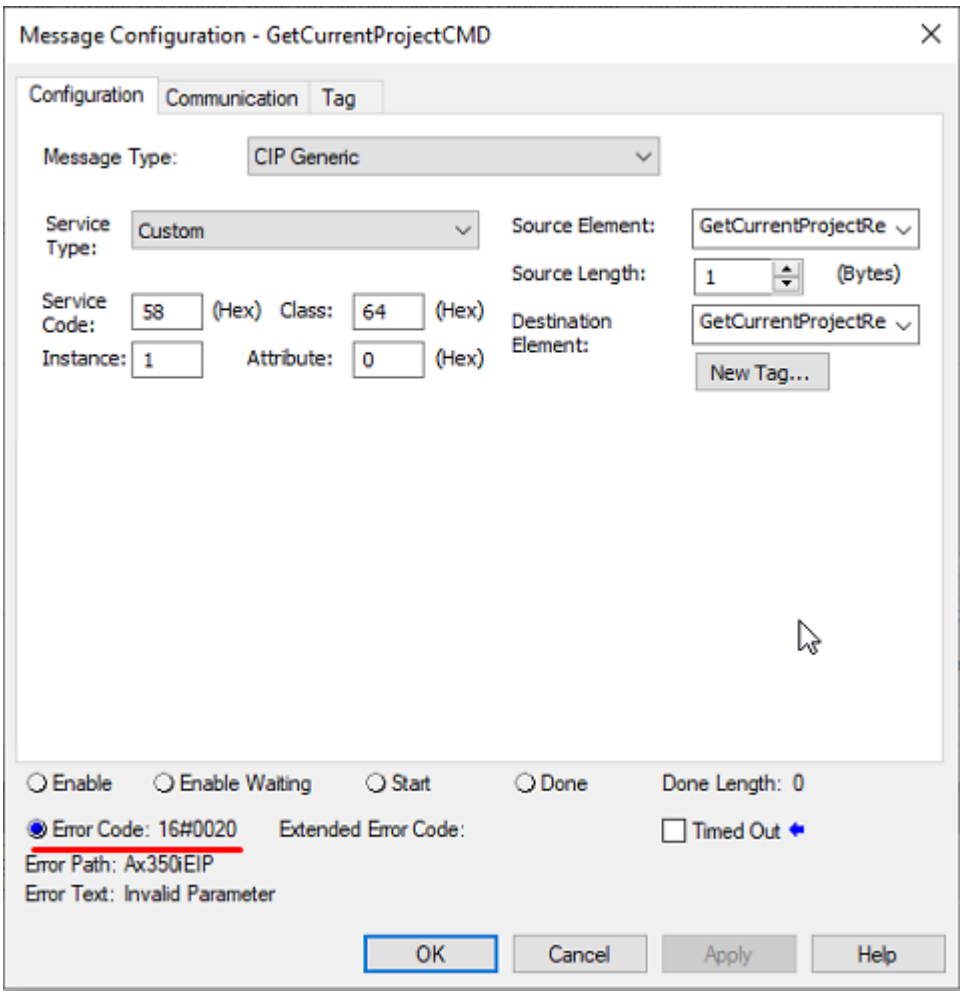


Figure 1: An error is returned. Studio 5000 screen copy of the MESSAGE CONFIGURATION screen in the Main Program.

5 APPENDIX A: Implicit Messages

5.1 Profile for D/F-Series

Product code = 0x1 **Product name** = "D/F-Series" This is if the EDS file for the Add-on profile from Domino is used.

The inbound information for D/F-Series are listed below:

Byte Position	Data Type	Parameter	Description
0	boolean	Printer Ready	Indicates whether the printer is in ready to print state.
1	boolean	Marking enabled	Indicates whether the marking mode is enabled.
2	boolean	Printing	Indicates when the printer is marking.
3	uint8_t	Severity	Indicates a change in the severity of the current print status. The severity published can be: • 0 = Information • 1 = warning • 2 = fault
4 to 7	uint32_t	Status	Indicates status change, the parameter contains the alert ID. For alert IDs reference the user guide of the marking machine.
8 to 11	uint32_t	Print Count	Publishes the number of prints done on the currently selected label.
12	boolean	Label Ready	Indicates when a label is selected to print.
13	boolean	Label content available	Indicates the content of the last print is available to read with an explicit command.
14	boolean	External data IDs available	Indicates the external data ID of the last print is available to read with an explicit command.

The outbound information for D/F-Series (sending from PLC to printer) are listed below:

Byte Position	Data Type	Parameter	Description
0	boolean	Enable/Disable Printing	Request to enable or disable printing. **Note:** The behaviour of this attribute is different to Ax-Series.
1	boolean	Trigger	Request a print.

5.2 Profile for Cx/Gx-Series

Product code = 0x2 **Product name** = "Gx-Series"

Product code = 0x3 **Product name** = "Cx-Series"

Gx and Cx have print heads which can be defined to belong into print groups. For each print group the individual implicit messages are generated. For example, Byte position 0 carries a Boolean showing if the Print Group 1 is ready. Print Groups that are not assigned will show a value of "0" in the respective byte.

The inbound information for Gx or Cx-Series are listed below:

Byte Position	Data Type	Parameter	Description
0 - 3	boolean	Ready to Print	System is turned on and marking is enabled. This message is also sent if new data has been compiled and set active because of external changes (e.g. SETTEXT); This is the equivalent to MSG1 in Dynamark. Byte 0 is Print Group1 through to Byte 3 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
4 - 7	boolean	Marking enabled	Indicates whether the marking mode is enabled; Equal with "Ready". Byte 4 is Print Group1 through to Byte 7 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
8 - 11	boolean	Printing	Indicates when the printer is marking. "0" is equivalent to MSG3 of Dynamark whilst "1" is equivalent MSG2 of Dynamark. Byte 8 is Print Group1 through to Byte 7 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
12	uint8_t	Severity	Indicates the severity of the current printer status for the printer as whole. The severity published can be: • 0 = Information • 1 = warning • 2 = fault
13 - 16	uint8_t	Severity	Indicates the severity of an individual Print Group. The severity published can be: • 0 = Information • 1 = warning • 2 = fault Byte 13 is Print Group1 through to Byte 16 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
17 to 20	Uint32_t	Status	Indicates the status of the printer as a whole. For status-IDs reference the product manual.
21 to 24 - 33 to 36	uint32_t	Status	Indicates the status of the Print Group. Byte 21 to 24 is Print Group1 through to Byte 33 to 36 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte. For status-IDs reference the product manual of the printer. Print Groups that are not assigned will show a value of "0" in the respective byte.

Byte Position	Data Type	Parameter	Description
37 to 40 - 49 to 52	uint32_t	Print Count	Publishes the number of prints done on the currently selected label (equivalent to GETPRINTCOUNTER). Byte 37-40 contain the value for Print Group 1 through to byte 49 to 52 containing the value for Group 4. Print Groups that are not assigned will show a value of "0" in the respective byte.
52 - 56	boolean	Label Ready	Indicates when a label is ready to print. This is the equivalent to MSG 26 in Dynamark. Byte 4 is Print Group1 through to Byte 7 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
57 - 60	boolean	Label content available	Indicates the content of the last print is available to read with an explicit command and that print has started. This is the equivalent to MSG24 in Dynamark. Byte 57 is Print Group1 through to Byte 60 being Print Group4. Print Groups that are not assigned will show a value of "0" in the respective byte.
61 - 64	boolean	Remote Coding Print	Indicates the print of a label has been completed. The external data ID of the last print is available to read with an explicit command. This is the equivalent of MSG25 in Dynamark.

The outbound information for Cx/Gx-Series (sending from LC to printer) are listed below:

Byte Position	Data Type	Parameter	Description
0 - 3	boolean	Enable Marking	Request to enable marking. Byte 0 is Print Group1 through to Byte 3 being Print Group4.
4 - 7	boolean	Stop Marking	Request to disable marking. Byte 4 is Print Group1 through to Byte 7 being Print Group4.
8 - 11	boolean	Trigger	Trigger to start marking a label. Byte 8 is Print Group1 through to Byte 11 being Print Group4.

6 APPENDIX B: Status (Error) Messages

Domino EtherNet/IP implementation defines and implements a set of General Status Codes following the CIP specification. The list of the supported General Status Codes along with their description can be found in Table 1.

Table 1: The table lists all CIP based General Status Codes used in the implementation of EtherNet/IP.

General Status Code (in hex)	Status Name	Description of status
05	Path destination unknown	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
08	Service not supported	The requested service was not implemented or was not defined for this Object Class/Instance.
13	Not enough data	The service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported.
15	Too much data	The service supplied more data than was expected.
16	Object does not exist	The object specified does not exist in the device.
1E	Embedded Service Error	There are multiple requests to the same message.
1F	Vendor specific error	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error. The Domino specific vendor error code list is in Table 2.
20	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.

For error code 0x1F Domino created a set of refined vendor specific error codes based on the error codes provided in the Dynamark protocol. These error codes are listed in Table 2.

Table 2: Extended Status Codes are returned along with General Status Code 0x1F. These are Domino specific messages based on the Dynamark protocol

Detailed Vendor Error Code	Description
1	No document loaded.
2	Wrong number of parameters.
3	Object with specified name not found.
4	Command unknown.
5	Wrong Object-Type.
6	Wrong Parameters.
7	Translation failed.
8	Specified Counter not found.
9	Error while File-I/O.
10	Timeout for a command that required a response.
11	No message must be open.
12	Source not found.
13	The function is not supported in this system configuration.
14	Internal fault.
15	The XML code that was sent is not valid.
16	Transaction is locked.
17	No transaction is open.
18	The variable does not exist in the current label.
19	Command parse error.
20	The index specified in the BUFFERDATA command is already in use or – if automatic id generation via id=-1 is used – more than 9999 data records have been preloaded.
21	Object cannot be added with OBJECTADD as the specified object name already exists.
22	Object creation failed (e.g. the specified object type is not supported).
23	Operation not allowed.
24	No connection to hardware module (only for D-Series).
25	No permissions to perform this action.
26	Remote data buffering is not active.
27	Cold start procedure failed (laser warm up incomplete).

Detailed Vendor Error Code	Description
28	Vector compilation failed (e.g. objects out of bounds or wrong encoder direction).
29	Internal Error occurring while logging in the communication interface user – please contact Domino.
30	Multi Printer configuration – The given printer name in a command doesn't exist.
31	Multi Printer configuration – The response of a controller doesn't match to the command.
32	Setting or getting network address or DNS is not successful.
33	Current command failed because the marking engine is still running. It can only work after the marking engine stops.

7 Command Reference

Note: Optional parameters are given in [ ] , while required parameters are given in < >.

7.1 Command Reference - Class Code 0x64

0x4B – BEGINTRANS

Command		
BEGINTRANS		
Description		
Starts a transaction. All the transaction commands following will be added to the transaction list instead of their immediate execution. The transaction list will be executed by sending the EXECTRANS command. Transactions are normally used for performance reasons, as the print project will not be updated while the commands of the list are received. This saves project preparation time and prohibits the printer to print half of the old data and half of the new if triggered while the update takes place. The result of BEGINTRANS can only be OK if no previously started transaction is still open (transactions are automatically closed and discarded if the connection gets lost).		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: “OK”.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
EXECTRANS		

0x4C – EXECTRANS

Command		
EXECTRANS		
Description		
Executes a transaction introduced with the BEGINTRANS command. Each transaction can only be executed once. Depending on the configuration of the printer, the external print go trigger may be disabled while executing the transaction (if "Force Recompile between Prints" is enabled manually or by automatic mode).		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: “OK”.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
BEGINTRANS		

0x4D – GETCOUNT

Command		
GETCOUNT <Length Countername><Countername>		
Description		
Reads the current reference value (Counter-ID) of the specified counter. This is always the internal numerical value of the counter.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Countername	Data Type: DINT Description: Size of <Countername>	
Countername	Data Type: UTF8 encoded array of SINTs of length <Length Countername>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Value>
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded Array of SINT of length <Length> returning the counter value
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETCOUNTERVALUE, SETCOUNTERVALUE, SETCOUNT		

0x4E - SETCOUNT

Command		
SETCOUNT <Length Counter Name><Counter Name><Length Counter Value><Counter Value>		
Description		
Sets the internal value for the specified <Counter Name> to the specified <Value>. The values set are non-formatted. To set formatted values SETCOUNTERVALUE must be used.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Counter Name	Data type: DINT Description: Size of <Counter Name>	
Counter Name	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Counter Value>	
Length Counter Value	Data type: DINT Description: Size of counter <Counter Value>	
Counter Value	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Counter Value>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault:	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		

See Also
SETCOUNTERVALUE, GETCOUNT, GETCOUNTERVALUE

0xF4 – GETCOUNTERVALUE

Command		
GETCOUNTERVALUE <Length Countername><Countername>		
Description		
Reads the current formatted value of the specified counter.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Countername	Data Type: DINT Description: Size of <Countername>	
Countername	Data Type: UTF8 encoded Array of SINT of length <Length Countername>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Value>
	Value	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length> returning the counter value
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
SETCOUNTERVALUE, SETCOUNT, GETCOUNT		

0x50 – SETCOUNTERVALUE

Command		
SETCOUNTERVALUE <Length Counter Name><Counter Name><Length Counter Value><Counter Value>		
Description		
Sets the internal value for the specified <Counter Name> to the specified formatted <Value>.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Counter Name	Data type: DINT Description: Size of <Counter Name>	
Counter Name	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Counter Value>	
Length Counter Value	Data type: DINT Description: Size of counter <Counter Value>	
Counter Value	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Counter Value>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault:	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Supported	
Gx-Series	Not Supported	
Examples		

See Also
GETCOUNTERVALUE, SETCOUNT, GETCOUNT

0x51 – GETPARSEDTEXT

Command		
GETPARSEDTEXT <Length Element><Element>		
Description		
Requests the text contents of the specified text or MRC object. Dynamic text contents like counters, clocks, inserts etc. will be parsed and returned as the current value. Notes: 1 Line breaks will be substituted with "<10>" as they are usually displayed within the properties list of the user interface. 2 This command will not return an automatically calculated checksum on MRC objects.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Element	Data Type: DINT Description: Size of <Element>	
Element	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Element>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Value>.
	Value	Data type: Array of SINT Description UTF8 encoded Array of SINT of length <Length>
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
SETTEXT, GETTEXT		

0x52 – GETTEXT

Command		
GETTEXT <Length><Element>		
Description		
Requests the text contents of the specified label element. Dynamic text content such as counters, clocks and inserts etc. will not return the dynamic values. These are returned with their token representation. Therefore the command GETTEXT shall only be used with static text. Notes: 1 Line breaks will be substituted with "<10>" as they are usually displayed within the property list of the user interface. 2 Only use on static text. To get the content from dynamic text such as counters, clocks and inserts etc. please use the command "GETPARSEDTEXT".		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Element	Data Type: DINT Description: Size of <Element>	
Element	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Element>	
Result Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Value>.
	Value	Data type: Array of SINT Description UTF8 encoded Array of SINT of length <Length>
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Supported	
Gx-Series	Supported	
Examples		

See Also
SETTEXT, GETPARSEDTEXT

0x53 – SETTEXT

Command		
SETTEXT<Length Element><Element><Length Value><Value>		
Description		
Assigns a new static text <Value> to a label element of name <Element>.		
Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Element	Data Type: DINT Description: Size of <Element>.	
Element	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Element>	
Length Value	Data Type: DINT Description: Size of <Value>.	
Value	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Value>	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		

See Also
GETTEXT

0x54 - GETVARLIST

Command		
GETVARLIST		
Description		
Returns a list <Variable List> of all variables existing in the label or the system (depending on the system configuration) If a variable contains a blank space all variables are put into “” (double quotes). The List of variables will be returned as multiple strings separated by a space character.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	GETVARLIST <Length><Variable list> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Variable list>.
	Variable list	Data type: Array of SINT Description UTF8 encoded Array of SINT of length <Length>
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not supported	
D-/F-Series	Supported	
Gx-Series	Not supported	
Examples		
Example: `"Text Variable 2" "Text Variable 3"  Text Variable`		
See Also		
GETVARIABLES, SETVARIABLES		

0x55 - GETVARIABLES

Command		
GETVARIABLES <Length> <Variable name>		
Description		
Returns the contents of the specified variable. If the variable does not error will be returned. If the string contains a blank space then “” will be transferred before and after the variable. The <Length> of the response string will then be enhanced by the 2 bytes.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Name	Data type: DINT Description: Size of <Variable name>.	
Variable name	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Name>	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Variable value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length Value	Data type: DINT Description: Size of <Variable value>.
	Variable value	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Value>. If the string contains a blank space there is “ (double quotes) before and after the text
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not supported	
D/F-Series	Supported	
Gx-Series	Not supported	
Examples		
See Also		
GETVARLIST, SETVARIABLES		

0x56 – SETVARIABLES

Command		
SETVARIABLES <Length><Variable name><Length><Variable value>		
Description		

Sets the contents of the specified variable.
The variables scope (local or global) is defined in the printer configuration.
Also, there are two behaviours for this command depending on the printer configuration:
• If the printer settings specify to allow to "create variables on access" (True):
  ○ If the variable with this <Variable name> ID does not exist, it will be newly created.
• If the printer settings specify to not allow to "create variables on access" (False):
  ○ If the variable does not exist, an error (Error-Code 18) will be returned.		

Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Name	Data Type: DINT Description: Size of <Variable name>.	
Variable name	Data Type: Array of SINT Description: Array of SINT of <Length Name> bytes of UTF8 encoded variable name.	
Length Variable Value	Data Type: DINT Description: Size of <Variable value>.	
Variable value	Data Type: Array of SINT Description: Array of SINT of 'Length Variable Value' bytes of UTF8 encoded content of the variable.	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	

Supported Technology	
Cx-Series	Not supported
D/F-Series	Supported
Gx-Series	Not supported
Examples	
See Also	
GETVARLIST, GETVARIABLES	

0x57 - LOADPROJECT

Command	
LOADPROJECT <Length Project ><Project>	
Description	
Loads the specified label for printing.	
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
Length Project	Data Type: DINT Description: Size of URl <Project>
Project	Data Type: Array of SINT Description: UTF8 encoded URI as Array of SINT of length <Length Project> This parameter references the label that shall be loaded. There are different formats of the URl available: store:<store ID>/<message ID>, specifying a store and label from where to load the label. If <store ID> is empty the currently selected message store is used (not all printers allow to select a current message store). "" (two double-quotes without a blank in between) for closing the current label “<filename>” Just a filename excluding the label extension to load a label from the currently selected message store (not all systems allow to select a current message store). For example: • “store:/label01” • “store:Store01/label01” “label01” For information about how the URI needs to be set, see the LOADPROJECT command in the “Dynamark Interface Communication Protocol User Guide” Doc-0018559
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
Success	<Length><Value> For details reference 4.1 Category A: “OK”.
	Length: Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value: Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.

Supported Technology	
Cx-Series	Supported
D/F-Series	Supported
Gx-Series	Supported
Examples	
See Also	
GETCURRENTPROJECT	

0x58 - GETCURRENTPROJECT

Command		
GETCURRENTPROJECT		
Description		
Determines the URI of the current label active for marking. The command returns an error '1' if no label has been loaded. If a new label has been created that is not yet saved or if the label was loaded with SETXML, the command returns an empty string.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><URI> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <URI>.
	URI	Data Type: Array of SINT Description: UTF8 encoded URI as Array of SINT of length <Length>.
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		
LOADPROJECT		

0x59 - SAVEPROJECT

Command	
SAVEPROJECT <Length URl><URI>	
Description	
Saves the current label using the filename specified as parameter.	
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
Data Type: DINT Description: Size of URl <Project>	
Data Type: Array of SINT Description: UTF8 encoded URI in an array of SINT of length <Length Project> This parameter references the label that shall be saved. There are different formats of the URl available as described below. Store:<store ID>/<message ID>, specifying a store and label from where to save the label. If <store ID> is empty the currently selected message store is used (not all printers allow to select a current message store).	
URI	This parameter saves the currently loaded label. There are different formats of the URI available: D-/F-Series only: • "file:<path>" With path being a path and filename or just a filename (the application then interprets this as relative to the message store root). • "store:<store ID>/<message ID>" Specifying a store and label from where to load the label. If <store ID> is empty the currently selected message store is used (not all systems allow to select a current message store). Since version 4.2.0.5 also allows to use a label extension. • "live:" Will store the information on the current live label to the internal storage device so it can be restored after power cut. • "<filename>" Just a filename excluding the label extension to save a label to the currently selected message store (not all systems allow to select a current message store) - Added since version 4.2.0.5 Gx only (< Version x.6.2.0 software): • "<filename>" Just a filename including the label extension • "live:" Will store the information on the current live label to the internal storage device so it can be restored after power cut.

URI	Gx only (>= Version x.6.2.0 software): • "file:<path>" With path being a path and filename or just a filename (the application then interprets this as relative to the label directory) • "store:<store ID>/<message ID>" Specifying a store and label to save the label. If <store ID> is empty the label store is used. Also allows to use a label extension. • "live:" Will store the information on the current live label to the internal storage device so it can be restored after power cut. • "<filename>" Just a filename excluding the label extension to save a label to the label directory. Cx only: • "file:<path>" With path being a path and filename or just a filename (the application then interprets this as relative to the label directory). • "store:<store ID>/<message ID>" Specifying a store and label to save the label. If <store ID> is empty the label store is used. Also allows to use a label extension. • "live:" Will store the information on the current live label to the internal storage device so it can be restored after power cut. • "<filename>" Just a filename excluding the label extension to save a label to the label directory.

Response Parameter Description
For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.

Success	<Length><Value> For details reference 4.1 Category A: "OK" Is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	

Supported Technology

Cx-Series	Supported
D/F-Series	Supported
Gx-Series	Supported Update in: From Version x.6.2.0

See Also

LOADPROJECT, GETCURRENTPROJECT

0x5A - GETPOSITIONOFFSET

Command		
GETPOSITIONOFFSET		
Description		
Requests the currently active x- and y-offsets in mm that are applied to the complete marking contents. Note: For Gx-Series, this command returns the current forward and end margin of the label loaded for printing.		
Result Description		
Success	<X-offset><Y-offset> For details reference 4.2 Category B: Numbers are Returned.	
	<X-offset>	Data Type: Float Description: Offset x-axis
	<Y-offset>	Data type: Float Description: Offset y-axis
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not supported	
D/F-Series	Supported	
Gx-Series	Supported For Gx-Series, this command returns the current forward and end margin of the label loaded for printing	
Examples		
See Also		

0x5B - SETPOSITIONOFFSET

Command		
SETPOSITIONOFFSET <X-offset><Y-offset>		
Description		
Sets the x- and y-offsets in mm that are applied to the complete marking contents. Note: In the Gx-Series, offset refers to forward margin and end margin. Attention: Be aware of the Dynamark coordinates. The lowest y-coordinate is on the top of the field.		
Parameter Description		
X-offset	Data Type: FLOAT Description: Offset x-axis	
Y-offset	Data Type: FLOAT Description: Offset y-axis	
Response Parameter Description		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		

0x5C – GETENCODERDELAY

Command		
GETENCODERDELAY <Global>		
Description		
Determines the current setting of the encoder delay in millimeters. • If global = 0, the encoder/print delay defined in the label will be returned. If the return value is empty (“”), the encoder/print delay is not defined in the label. • If global = 1, the global encoder/print delay will be returned.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Global	Data Type: BOOL Description: • If global = 0, the encoder/print delay defined in the label will be returned. If the return value is empty (“”), the encoder/print delay is not defined in the label. • If global = 1, the global encoder/print delay will be returned.	
Response Parameter Description **For detailed description reference . 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT	
	Length	Data Type: DINT Description: Size of <Value>
	Value	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of <Length> byte length If the return value is empty (“”), the encoder/print delay is not defined in the label. The Array of SINT returns the delay value as a float with two decimals. The decimals are discriminated by a “.”. An example would be “4.99” or “5.00”
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	

Examples
See Also
SETENCODERDELAY

0x5D – SETENCODERDELAY

Command		
SETENCODERDELAY <Global><Length Delay><Delay>		
Description		
This command enables modifying the encoder/product detect delay in millimetres.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Global	Data Type: BOOL Description: • If global = 0, the encoder delay defined in the label will be set. • If global = 1, the encoder delay will be set globally in the printer (printer delay).	
Length Delay	Data Type: DINT Description: Size of <Delay>.	
Delay	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Delay>. The delay value is required to be a number with two digits. The decimal discriminator must be a ".". For example: "2.00"	
Result Description		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETENCODERDELAY		

0x5E – RESETSYSTEM

Command		
RESETSYSTEM		
Description		
This command resets pending warning and faults.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference4.1 Category A: “OK”.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		

0x5F – GETDATA

Command		
GETDATA <Object Length> <Object> <Data-ID>		
Description		
Reads the specified property attribute from the specified object. The numeric identifier of each property can be obtained in 8 APPENDIX C: Data-IDs for SETDATA/GETDATA.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Object Length	Data Type: DINT Description: Size of <Object>.	
Object	Data Type: Array of SINT Description: SINT array of length <Object Length> With UTF8 encoded Object name.	
Data-ID Length	Data Type: DINT Description: Size of <Data-ID>	
Data-ID	Data Type: Array of SINT Description: SINT array of length <Data-ID Length> With UTF8 encoded Data-ID. Identifier as in Appendix	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length. <Response>	
	Value Length	Data Type: DINT Description: Length of <Value>
	Value	Data type: Array of SINT array of SINT of length <Length>
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not supported	
D/F-Series	Supported	
Gx-Series	Not supported	
Examples		

See Also
SETDATA, APPENDIX C: Data-IDs for SETDATA/GETDATA

0x60 – SETDATA

Command		
SETDATA <Object Length> <Object> <Data-ID Length><Data-ID> <Data Length><Data>		
Description		
Sets value of attribute specified by Data-ID. The numeric identifier of each property can be obtained in 8 APPENDIX C: Data-IDs for SETDATA/GETDATA.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Object Length	Data Type: DINT Description: Size of <Object>.	
Object	Data Type: Array of SINT Description: SINT array of length <Object Length> With UTF8 encoded Object name.	
Data-ID Length	Data Type: DINT Description: Size of <Data-ID>.	
Data-ID	Data Type: : Array of SINT Description: SINT array of length <Data-ID Length> With UTF8 encoded Data-ID. Description: Identifier as in Appendix	
Data Length	Data Type: DINT Description: Size of <Data>.	
Data	Data Type: Array of SINT Description: SINT array of length <Data Length> With UTF8 encoded data.	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length. <Response> For details reference 4.1 Category A: "OK".	
	Value Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT Description: Array of SINT of 'Value Length' bytes of UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	

Supported Technology	
Cx-Series	Not supported
D-/F-Series	Supported
Gx-Series	Not supported
Examples	
See Also	
GETDATA, APPENDIX C: Data-IDs for SETDATA/GETDATA	

0x61 - GETVERSION

Command		
GETVERSION		
Description		
This command obtains the version of the printer controller software and the versions of soft- and hardware of the marking controller and the version of the controller image.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length Software><Software><Length DSP><DSP><Length Image><Image>	
	Length Software	Data Type: DINT Description: Size of <Software>.
	Software	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Software>.
	Length DSP	Data Type: DINT Description: Size of <DSP>.
	DSP	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length DSP>.
	Length Image	Data Type: DINT Description: Size of <Image>.
	Image	Data Type: Array of SINT Description: UTF8 encoded Array of SINT of length <Length Image>.
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		

0x63 – GENERIC_DYN

Command		
GENERIC_DYN <Length Command><Command>		
Description		
This command sends a Dynamark command as an Array of SINT. This is a method to overcome any functionality that is not exposed through the EtherNet/IP interface. Note: Because the error messages generated by Dynamark are the basis for the error codes returned in this EtherNet/IP interface, the error is returned if a fault happens. This is different to the GENERIC_CODENET command.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Length Command	Data Type: DINT Description: Size of the Dynamark <Command>	
Command	Data Type: Array of SINT Description: Array of SINT of length 'Length Command'. Note: Example: 'SETTEXT "TEXT 1" "Halmackenreuther"' will be the entry of <Command>	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data Type: DINT Description: Size of <Value>.
	Value	Data Type: Array of SINT Description: UTF8 Array of SINT of length <Length>.
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		

7.2 Command Reference - Class Code 0x65

0x4B – BUFFERCLEAR

Command		
BUFFERCLEAR		
Description		
This command clears all records from the remote queue.		
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: “OK” is returned.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		
BUFFERDATA, BUFFERREVOKE, GETBUFFERSTATUS, BUFFERCONFIG, BUFFERDATAALLGROUPS, BUFFERCLEARALLGROUPS		

0x4C – BUFFERDATA

Command		
BUFFERDATA <ID><Length Argument><Argument>		
Description		
This command inserts a record into the remote queue. For synchronization purposes, the record can be provided with a unique numeric ID that will be returned by the messages 25. If not required, the ID can be set to "-1" to automatically generate these IDs internally by the system. If you use the automatic ID generation (ID=-1) there is a limitation of 9999 records that can be preloaded. Note: To avoid conflicts and difficulties in printer behaviour the BUFFERDATA function has been limited to one argument at a time. This is different from the function in Dynamark.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
ID	Data Type: DINT Description: ID number of the buffer value	
Length Argument	Data Type: DINT Description: length of the Array of SINT <Argument>	
Argument	Data Type: Array of SINT Description: Argument to be inserted in the buffer at ID position. Length of the Array of SINT is defined in <Length Argument>	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Supported	
Gx-Series	Supported	
Examples		

See Also
BUFFERREVOKE, GETBUFFERSTATUS, BUFFERCLEAR, BUFFERCONFIG, BUFFERDATAALLGROUPS, BUFFERCLEARALLGROUPS

0x4D – BUFFERREVOKE

Command	
BUFFERREVOKE<Parameter Name Length><Parameter Name><Parameter Value>	
Description	
This command deletes an entry or multiple entries from the remote queue in the way specified by <Parameter Name>.	
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
Parameter Name Length	Data Type: DINT Description: Size of <Parameter Name>
Parameter Name	Data type: Array of SINT Description: <Parameter Name> can have two values: • 'record': Discard the record <Parameter Value> from the buffer list • 'maximum_record_age': Discard all records older than a maximum age in seconds specified by <Parameter Value>.
Parameter Value	Data Type: DINT Description: Value the parameter shall have.
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.
	Length: Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value: Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.
Supported Technology	
Cx-Series	Not Supported
D/F-Series	Supported
Gx-Series	Not Supported

Examples

Example: This command erases buffer entry ID 1 in the buffer list:
`BUFFERREVOKE$06$00$00$00record$01$00$00$00`
 	
Example: This command erases all buffer entries older than 2 seconds:
`BUFFEREVOKE$12$00$00$00maximum_record_age$02$00$00$00`

See Also
BUFFERDATA, BUFFERCLEAR, GETBUFFERSTATUS, BUFFERCONFIG, BUFFERDATAALLGROUPS, BUFFERCLEARALLGROUPS

0x4E – GETBUFFERSTATUS

Command		
GETBUFFERSTATUS		
Description		
Use of this command obtains the number of data records in memory of the marking system controller still to be printed.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Level> For details reference 4.2 Category B: Numbers are Returned.	
	Level	Data type: DINT Description: Buffer level
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		
BUFFERDATA, BUFFERCLEAR, BUFFERREVOKE, BUFFERCONFIG, BUFFERCLEARALLGROUPS, BUFFERDATAALLGROUPS BUFFERDATA, BUFFERCLEAR, BUFFERREVOKE, BUFFERCONFIG, BUFFERDATAALLGROUPS, BUFFERCLEARALLGROUPS		

0x4F – GETCODINGFILE

Command		
GETCODINGFILE		
Description		
This command returns the <File Name> of the coding file for the currently selected message and the <Number of Codes> that are used for one label print. Notes: 1 The file coding feature must already be turned on. 2 Remote data coding will return an empty Array of SINT as parameter.		
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<File Name Length><File Name><Number of Codes> For details reference 4.2 Category B: Numbers are Returned & 4.3 Category C: Content is Returned as Array of SINT.	
	File Name Length	Data Type: DINT Description: Size of < File Name>
	File Name	Data type: Array of SINT Description: UTF8 encoded file name
	Number of Codes	Data Type: DINT Description: Number of codes being used per label
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETCODINGLINE, SETCODINGLINE, SETCODINGFILE		

0x50 – SETCODINGFILE

Command		
SETCODINGFILE<File Name Length><File Name> [Number of Codes Length][Number of Codes]		
Description		
This command sets the <File Name> of the coding file for the currently loaded label and optionally the [Number of Codes] lines to be read per print. Note: The coding feature must already be configured to file mode in the label settings. Attention: The system must not be in marking mode when this command is issued.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
File Name Length	Data Type: DINT Description: Size of <Parameter Name>	
File Name	Data type: Array of SINT Description: UTF8 encoded file name as Array of SINT of length <File Name Length>	
Number of Codes Length	Data Type: DINT Description: Optional Parameter that must be present if Number of Codes shall be given. Size of [Number of Codes]	
Number of Codes	Data Type: Array of SINT Description: Optional Parameter that requires also [Number of Codes Length] to be present if used. Number of codes being used per label.	
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	

Examples
if file name is “demo.txt” and number of code lines per label is 10: `$09$00$00$00demo.txt$0A$00$00$00`
See Also
GETCODINGLINE, GETCODINGFILE, SETCODINGLINE

0x51 – GETCODINGLINE

Command		
GETCODINGLINE		
Description		
This command returns the current line number in the file that is used for file coding.		
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Line Number> For details reference 4.2 Category B: Numbers are Returned.	
	Line Number	Data Type: DINT Description: Line number used to read the information for print; little endian
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
A line number of 10 is represented as: $0A$00$00$00		
See Also		
SETCODINGLINE, GETCODINGFILE, SETCODINGFILE		

0x52 – SETCODINGLINE

Command		
SETCODINGLINE<Line Number>		
Description		
Use of this command sets the current line number in the file that is used for file coding. Note: Using this command the customer needs to take care not to print parts of a coding file multiple times.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Line Number	Data Type: DINT Description: Line Number	
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETCODINGLINE, GETCODINGFILE, SETCODINGFILE		

0x53 – GETMSG24

Command		
GETMSG24		
Description		
This command queries for MSG24. This message is triggered on every print start providing the contents of all text/MRC objects in the label. The contents are always provided in an object name-content pair but are encoded in a single ARRAY of SINT per label. Both object name and contents are quoted (using “<contents>”) if spaces are included, using preceding backslash for characters like “, linefeed, newline, tab. The content of the label can be queried until the next label starts printing. The command is designed to work with the implicit message “Label Content Available”. See 5 APPENDIX A: Implicit Messages. The content of the last printed labels. The content of up to 10 labels is buffered in a FIFO buffer.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length of MSG24>< MSG24> For details reference 4.1 Category A: “OK” is Returned.	
	Length of MSG24	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	MSG24	Data type: Array of SINT Description: UTF8 encoded content of MSG24
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		
GETMSG25		

0x54 – GETMSG25

Command		
GETMSG25		
Description		
This command can be used in case remote data coding is active and a print has been triggered and printed completely. The ID that is returned is the ID used with the BUFFERDATA command of the record that just has been printed successfully. The message will not return the ID if coding is interrupted, e.g. by opening an interlock door (because the label is not completely coded). The ID of the label content can be queried until the next label with new content starts printing. The command is designed to work with the implicit message "Label Content Available". See 5 APPENDIX A: Implicit Messages.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length of MSG25>< MSG25> For details reference 4.1 Category A: "OK" is Returned.	
	Length of MSG24	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	MSG24	Data type: Array of SINT Description: UTF8 encoded content of MSG25
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		
GETMSG24		

0x55 - GETSYSTEMCOUNTER

Command		
GETSYSTEMCOUNTER <Type>		
Description		
This command returns the current value of a series of counters depending on the printer technology. Note: The type is being returned along with its value. Both little endian.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Counter type	Data type: DINT Description: Counter type that is requested: D-Series: • 1 = Runtime of coder in seconds. • 2 = Runtime of laser tube in seconds. • 3 = Number of prints for currently selected label. This counter will rest when reloading a label or after restarting the laser coder. • 4 = Overall number of prints (not resettable). • 5 = Number of prints since last reset (resettable to user-defined start value). Ax-Series: • 1 = Machine time in seconds • 2 = Jet time in seconds. • 3 = Number of prints for currently selected label. This counter is always associated to the label and therefore will not reset. • 4 = Overall number of prints (not resettable). • 5 = Number of prints since last reset (resettable to user-defined start value).	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Type><Counter> For details reference 4.2 Category B: Numbers are Returned.	
	Type	Data type: DINT Description: Counter type as described in the input above. Little endian
	Counter	Data type: DINT Description: Value of the counter indicated by counter type. Little endian
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	

Supported Technology	
Cx-Series	Not Supported
D/F-Series	Supported
Gx-Series	Not Supported
Examples	
See Also	
SETSYSTEMCOUNTER	

0x56 – SETSYSTEMCOUNTER

Command		
SETSYSTEMCOUNTER <Counter type><Counter value>		
Description		
This command returns the current value of a series of counters depending on the printer technology.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Counter type	Data type: DINT Description: Counter type that is requested D-Series: • 1 = Runtime of coder in seconds (requires service dongle). • 2 = Runtime of laser tube in seconds (requires service dongle). • 4 = Overall number of prints (requires service dongle). • 5 = Number of prints since last reset (resettable to user-defined start value). Ax-Series: • 5 = Number of prints since last reset (resettable to user-defined start value).	
Counter value	Data type: DINT Description: New value to set the counter type. Little endian	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: “OK” is Returned.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Supported	
Gx-Series	Not Supported	
Examples		

See Also
GETSYSTEMCOUNTER

0x57 – GETPRINTCOUNTER

Command		
GETPRINTCOUNTER		
Description		
This command returns the number of prints since the printer transitioned to ready to print. Note: This counter is reset when reaching ready to print.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Counter> For details reference 4.2 Category B: Numbers are Returned.	
	Counter	Data Type: DINT Description: Value of the print counter little endian.
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	
Examples		
See Also		

0x58 – SETMAXCOUNTS

Command		
SETMAXCOUNTS<Number of Prints>		
Description		
This command limits the <number of prints> after the printer reaches the ready to print state to a maximum.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Number of Prints	Data Type: DINT Description: Maximum Number of Prints	
Response Parameter Description: **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETCODINGLINE, GETCODINGFILE, SETCODINGFILE, SETCODINGLINE		

0x59 – GETCO2ONPULSEOFFSET

Command		
GETCO2ONPULSEOFFSET		
Description		
This command requests the current system global offset for the CO2 laser on pulse (absolute laser power offset)		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Offset> For details reference 4.2 Category B: Numbers are Returned.	
	Offset	Data Type: DINT Description: Value of the Offset
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Supported	
Gx-Series	Not Supported	
Examples		

The return of an offset of 10 is represented as:
`$0a$00$00$00`
 	
The return of an offset of -10 is represented as:
`$f6$ff$ff$ff`		

See Also		

0x5A – GETDATE

Command		
GETDATE		
Description		
Returns the current time and date of the printer/marking system.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Hour><Minute><Second><Year><Month><Day> For details reference 4.2 Category B: Numbers are Returned.	
	Hour	Data type: DINT Description: Hour as 2 digits with leading zeros. Little endian
	Minute	Data type: DINT Description: Minute as 2 digits with leading zeros. Little endian
	Second	Data type: DINT Description: Second as 2 digits with leading zeros. Little endian
	Year	Data type: DINT Description: Year as 4 digits with leading zeros. Little endian
	Month	Data type: DINT Description: Month as 2 digits with leading zeros. Little endian
	Day	Data type: DINT Description: Day as 2 digits with leading zeros. Little endian
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Supported	
Gx-Series	Supported	

Examples
Result could be: `\x0f\x00\x00\x00\x15\x00\x00\x00\x13\x00\x00\x00\xe7\x07\x00\x00\x02\x00\x00\x00\n\x00\x00\x00` With: `\x0f\x00\x00\x00` --> 15 hour `\x15\x00\x00\x00` --> 21 min `\x13\x00\x00\x00` --> 19 sec `\xe7\x07\x00\x00` -->2023 `\x02\x00\x00\x00` --> 2 - Feb `\n\x00\x00\x00` --> 10 day
See Also
SETTIME, SETDATE

0x5B – SETTIME

Command		
SETTIME <Hour><Minute><Second>		
Description		
Set the time of the printer/marking system.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Hour	Data type: DINT as little endian Description: Hour	
Minute	Data type: DINT as little endian Description: Minute	
Second	Data type: DINT as little endian Description: Second	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D/F-Series	Supported	
Gx-Series	Not Supported	
Examples		
See Also		
GETDATE, SETDATE		

0x5C – SETDATE

Command		
SETDATE <Hour><Minute><Second><Year><Month><Day>		
Description		
Set the current time and date of the marking system controller.		
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Hour	Data type: DINT as little endian Description: Hour	
Minute	Data type: DINT as little endian Description: Minute	
Second	Data type: DINT as little endian Description: Second	
Year	Data type: DINT as little endian Description: Year	
Month	Data type: DINT as little endian Description: Month	
Day	Data type: DINT as little endian Description: Day	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: “OK” is Returned.	
	Length	Data Type: DINT Description: Size of standard “OK” response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded “OK”: $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Supported	
Gx-Series	Supported	

Examples
See Also
GETDATE, SETTIME

7.3 Command Reference - Class Code 0x66

0x4B - BUFFERCLEARALLGROUPS

Command		
BUFFERCLEARALLGROUPS		
Description		
This command clears all records from the remote queue (filled with the BUFFERDATA or BUFFERDATAALLGROUPS command), in all configured print groups.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
See Also		
BUFFERDATA, BUFFERDATAALLGROUPS		

0x4C – BUFFERCONFIG

Command		
BUFFERCONFIG <Parameter Length> <Parameter>		
Description		
This command configures the remote queue. The options supported are: • maximum_record_age: The maximum time in seconds before a record is discarded. Set it to -1 if you don't want the records to expire (default).		
Request Parameter Description		
Parameter Length	Data type: DINT Description: Size of <Parameter value>	
Parameter	Data type: Array of SINT Description: UTF8 encoded text of length <Parameter Length>. For example: • "maximum_record_age 1" → Set the life of the entry to 1 second • "maximum_record_age -1" → Disable expiration of the entries	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Not Supported	
Gx-Series	Not Supported	
Examples		
See Also		

0x4D – BUFFERDATAALLGROUPS

Command	
BUFFERDATAALLGROUPS <ID> <Length1> <Arg1> <Length2> <Arg2> <Length3> <Arg3> <Length4> <Arg4>	
Description	
This command works in the same manner as the BUFFERDATA command, except the data is sent to any print group where a CObjVarText (see GETOBJECTS command) object is present.	
Request Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**	
ID	Data type: DINT Description: Data ID - Unique identifier of the data record. Set to -1 if not required.
Length1	Data type: DINT Description: Size of Arg1
Arg1	Data type: SINT[] Description: 'Length1' bytes of UTF8 encoded text
Length2	Data type: DINT Description: Size of Arg2
Arg2	Data type: SINT[] Description: 'Length2' bytes of UTF8 encoded text
Length3	Data type: DINT Description: Size of Arg3
Arg3	Data type: SINT[] Description: 'Length3' bytes of UTF8 encoded text
Length4	Data type: DINT Description: Size of Arg4
Arg4	Data type: SINT[] Description: 'Length4' bytes of UTF8 encoded text

Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Not Supported	
D-/F-Series	Not Supported	
Gx-Series	Not Supported	
Examples		
See Also		
BUFFERDATA		

0x4F - GETPROJECTS

Command		
GETPROJECTS		
Description		
Returns all labels that are stored in the current message store. Multiple labels come in a sequence separated by a space. If any label name has a space then " " are placed around the label name.		
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	GETPROJECTS <Length> <Name> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length	Data type: DINT Description: Size of labels
	Name	Data type: Array of SINT Description: 'Length' bytes of UTF8 encoded labels. For example, "label01" "label02" "label03"
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D-/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
If you have 3 labels: Demo, Empty, and test with space , the return is: `Demo Empty "test with space"`		
See Also		

0x50 - GETSTATUS

Command		
GETSTATUS <all-messages>		
Description		
Determines the status of the controller (message with the highest priority) and returns the translated text, the status-id and the describing text in the selected translation.		
Request Parameter Descriptions		
all-messages	Data type: BOOL Description: Parameter to retrieve the information on all active alerts and status. If the parameter is set to 1 then the alerts are returned in addition to the current status. If the parameter is set to 0 only the current status is returned when no other alerts are raised, otherwise, the highest severity alert is returned.	
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Number of alerts> <Severity> <Alert/status ID> <Length of alert/status text> <Alert/status text> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Number of alerts	Data type: DINT Description: Number of alerts if all-messages is 1 or 0 if no alerts. If all-messages is omitted or 0 then this argument is not present.
	Severity	Data type: DINT Severity: • 0=information • 1=warning • 2=temporary fault • 3=critical fault • 4=critical fault (needs to be reset by hardware)
	Alert/status ID	Data type: DINT Description: See operations manual for each product.
	Length of alert/status text	Data type: DINT Description: alert/status text returned in the language that is currently selected on the user-interface.
	Alert/status text	Data type: Array of SINT Description: UTF8 encoded alert/status text.

Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.
Supported Technology	
Cx-Series	Supported
D/F-Series	Not Supported
Gx-Series	Supported
Examples	
See Also	

0x51 - GETTOTALPRINTCOUNTER

Command		
GETTOTALPRINTCOUNTER		
Description		
Obtains the total print count since the last user reset.		
Response Parameter Descriptions **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length of total print counters><Current total print counter> For details reference 4.3 Category C: Content is Returned as Array of SINT.	
	Length of total print counters	Data type: DINT Description: Size of labels
	Current total print counter	Data type: Array of SINT Description: UTF8 encoded Array of SINT of length 'Length of total print counters'
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
See Also		
RESETTOTALPRINTCOUNTER		

0x52 - RESETTOTALPRINTCOUNTER

Command		
RESETTOTALPRINTCOUNTER		
Description		
Reset total print counter to zero.		
Response Parameter Descriptions **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK".	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
See Also		
GETTOTALPRINTCOUINTER		

0x54 – SETPARAM

Command		
SETPARAM <Parameter Length 1> <Parameter 1> [Parameter Length 2] [Parameter 2] <Value Length> <Value>		
Description		
This command is used for setting setup and runtime parameters of the printer. Only one parameter can be changed per call of the function.		
Request Service Data Field Parameter Descriptions **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Parameter Length 1	Data type: DINT Description: Size of <Parameter 1>	
Parameter 1	Data type: Array of SINT Description: UTF8 encoded parameter Array of SINT. In case more than one Array of SINT are necessary to describe the parameter the optional [Parameter 2] Array of SINT will be proceeded by its length will be required.	
Parameter Length 2	Data type: DINT Description: Size of <Parameter 2>	
Parameter 2	Data type: Array of SINT Description: UTF8 encoded parameter Array of SINT. This Array of SINT is only required if the parameter description requires more than one Array of SINT to contain all details.	
Value Length	Data type: DINT Description: Size of <Value>	
Value		
	Parameter Description – Gx-Series	
	SelectedGroup	Number of selected print group [1 to 4]. *Depends on model type.
	VariableTrigger	Sets customer specific GPIO-Output Delay distance [0 to 2500 mm] of a print group [1 to 4], triggered by Product Detect signal. *Depends on model type. Note: The print group must be selected with the SELECTGROUP command prior to this function.
	Parameter Description – Cx-Series	
	SelectedGroup	Selects the number of a print group [1 to 4]. *Depends on model type.
	VariableTrigger	Sets customer specific GPIO-Output Delay distance [0 to 2500 mm] of a print group [1 to 4], triggered by Product Detect signal. *Depends on model type.

Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	<Length><Value> For details reference 4.1 Category A: "OK" is Returned.	
	Length	Data Type: DINT Description: Size of standard "OK" response: $02$00$00$00
	Value	Data type: Array of SINT of length <Length> Description: UTF8 encoded "OK": $4f$4b
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
See Also		
GETPARAM		

0x55 – GETPARAM

Command		
GETPARAM <Parameter Length 1> <Parameter 1> [Parameter Length 2] [Parameter 2]		
Description		
This command is used for getting setup and runtime parameters of the printer. Only one parameter can be requested per function call.		
Request Service Data Field Parameter Descriptions **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Parameter Length 1	Data type: DINT Description: Size of <Parameter 1>	
Parameter 1	Data type: Array of SINT Description: UTF8 encoded parameter Array of SINT. In case more than one Array of SINT are necessary to describe the parameter the optional [Parameter 2] Array of SINT will be proceeded by its length will be required.	
Parameter Length 2	Data type: DINT Description: Size of <Parameter 2>	
Parameter 2	Data type: Array of SINT Description: UTF8 encoded parameter Array of SINT. This Array of SINT is only required if the parameter description requires more than one Array of SINT to contain all details.	
	Parameter Description – Cx-Series	
	InkLevel	Ink level of reservoir & ink bottle [0.0 ml to Max Reservoir & Max Bottle volume ml].
	NumberOfHeads	Number of print heads [1 to 4] as SINT. *Depends on model type.
	SelectedGroup	Number of selected print group [1 to 4] as SINT. *Depends on model type.
	NumberOfGroups	Number of print groups [1 to 4] as SINT. *Depends on model type.
	VariableTrigger <print group>	Customer specific GPIO-Output Delay distance [0 to 2500 mm] of a print group [1 to 4], triggered by Product Detect signal. *Depends on model type.
	SerialnoController	Serial number of the controller. Returns the serial number as Array of SINT preceded by the length of the Array of SINT as DINT. Returns “” if no serial number is present.

	Parameter Description – Gx-Series	
	InkLevel n	(1 - Standard): Ink level of cartridge in head n [0.0 ml to 42.0 ml]. (2 - Bulk): Life parameter of cartridge in head n [0.0 ml to 1210.0 ml].
	NumberOfHeads	Number of print heads [1 to 4] as SINT. *Depends on model type.
	SelectedGroup	Number of selected print group [1 to 4] as SINT. *Depends on model type.
	NumberOfGroups	Number of print groups [1 to 4] as SINT. *Depends on model type.
	VariableTrigger <print group>	Customer specific GPIO-Output Delay distance [0 to 2500 mm] of a print group [1 to 4], triggered by Product Detect signal. *Depends on model type.
	SerialnoController	Serial number of the laser controller. Returns the serial number as Array of SINT preceded by the length of the Array of SINT as DINT.
Response Parameter Description **For detailed description reference 1.6 Lengths and UTF8 Encoded Strings and Numbers.**		
Success	The returns are described in the "Request Service Data Field Parameter Descriptions" section.	
Fault	<code> For details reference 6 APPENDIX B: Status (Error) Messages.	
Supported Technology		
Cx-Series	Supported	
D/F-Series	Not Supported	
Gx-Series	Supported	
Examples		
See Also		
SETPARAM		

8 APPENDIX C: Data-IDs for SETDATA/GETDATA

DataID	Object Type	Description	D/F-Series Support	Cx/G/Gx-Series Support
2	All	Parameter set name	x	
3	All	Object is marked (0=no; 1=yes; 2=based on digital inputs)	x	
10	All	x-Position [mm]	x	
11	All	y-Position [mm]	x	
12	All	Width [mm]	x	
13	All	Height [mm]	x	
14	All	Rotation [°]	x	
15	All	Tip [°]	x	
16	All	x-Center (0=left; 1=center; 2=right; 3=none)	x	
17	All	y-Center (0=top; 1=center; 2=bottom; 3=none)	x	
18	All	x-Mirror (0=no; 1=yes)	x	
19	All	y-Mirror (0=no; 1=yes)	x	
20	All	IO Condition Mask	x	
21	Arc	IO Condition Value	x	
30	Arc	Angle [°]	x	
31	Arc, DXF	Secants	x	
32	Arc, Rectangle	Border Width	x	
33	Arc, Rectangle	Number of borders	x	
34	DXF	Join Lines (0=no; 1=yes)	x	
35	DXF	Tolerance for Join Lines	x	
50	Text, MRC	Text	x	
51	Text	Font	x	
52	Text	Spatio	x	
53	Text	Line Feed	x	
55	Text	Italics y [°]	x	
56	Text	Proportional (0=no; 1=yes)	x	
57	Text	Alignment (0=left; 1=center;2=right)	x	
58	Text	Radius for Rotation Mode	x	
59	Text	Rotation Mode (0=no; 1=clockwise; 2=counter-clockwise)	x	
60	Bitmap, Hpgl, DXF	Filename (incl. Path)	x	

DataID	Object Type	Description	D/F-Series Support	Cx/G/Gx-Series Support
61	Bitmap	Resolution [dpi]	x	
62	Bitmap	Algorithm (0=horizontal, 1=vertical, 2=outline)	x	
74	MRC	QR Code Version	x	
75	MRC	QR Code Format	x	
76	MRC	QR Code EC Level	x	
77	MRC	QR Code Mask	x	
78	MRC	Datamatrix Format	x	
79	MRC	MRC Check Digit	x	
80	MRC	Codetype (see Appendix)	x	
82	MRC	Resolution [dpi]	x	
83	MRC	x-Quietzone [mm]	x	
84	MRC	y- Quietzone [mm]	x	
90	MRC	Datamatrix Start Mode	x	
91	MRC, Bitmap	Invers (0=no; 1=yes)	x	
93	MRC	Security Level	x	
94	MRC	Algorithm (0=lines; 1=circles)	x	
95	MRC	Diameters per Cell	x	
96	MRC	Secants	x	
97	MRC	RSS Code Start Mode	x	
98	MRC	Datamatrix Square Size	x	
99	MRC	Datamatrix Rectangular Size	x	
100	Text	Inverse Lineflow (0=no; 1=yes)	x	
101	Text	Alternating Lineflow (0=no; 1=yes)	x	
102	Text	OCR-Link	x	
103	MRC	PDF 417 EC-Level	x	
104	MRC	PDF 417 Number of rows	x	
105	MRC	PDF 417 Number of columns	x	
106	MRC	PDF 417 Ratio	x	
107	MRC	QR Code Application Identifier	x	
108	MRC	QR Code Format	x	
109	MRC	QR Code Number of rows	x	
110	MRC	QR Code Number of columns	x	
120	Bitmap, HPGL, DXF	Force automatic load of file when file has been changed	x	
124	MRC	2D Line flow	x	
125	MRC	Short bar length	x	
126	MRC	Composite Code	x	
127	MRC	Part of the MRC to be marked	x	
128	MRC	Quiet Zone algorithm	x	

DataI D	Object Type	Description	D/F-Series Support	Cx/G/Gx-Series Support
129	All	Position data for object's bounding frame (left, top, right, bottom) in mm (read only)	x	

9 Document Reference

Doc-0023889 User Guide: EtherNet/IP Interface Communication Protocol Using a Domino Interface Controller
Doc-0018559 User Guide: Dynamark Interface Communication Protocol