Domino Product Group
G-Series TIJ Products
Dynamark Protocol Manual Issue1

Dynamark Protocol for G-Series

Manual

Document Reference Number:	DOC-000xxxx
Document Revision:	1,0
Document Status:	Issue 1
Document Issued Date:	November 2013

Domino UK Ltd. Bar Hill Cambridge CB23 8TU England

Modification Record

Revision	Date	Author	Changes (including Change Management reference)	Approval by
1.0	18th September 2013	Michael Koegler	First version	
1.1	18. October 2013	MK	Command echo is not supported. Add configuration parameter turning print group number appendage on/off. GETPROJECTS: exchanged "project" by "filename". GETPROJECTS: corrected type to non-print-group-command. SETXML/GETXML: added restrictions and comments. SETXML/GETXML: corrected project version number, correct XML data LOADPROJECT/SAVEPROJECT: removed identifier "file:". SAVEPROJECT: added comments and example SETDATE: change syntax for optional date arguments.	

Contents

1. Introduction

The Dynamark protocol is implemented in Domino Laser and will be implemented in the G-Series end of 2013. This document describes the protocol commands that will be implemented into G-Series. A succeeding document will be created in due time, which will show the protocol commands for Laser and G-Series in one common document.

2. Dynamark Protocol Overview

This document describes the commands implemented in the Domino G-Series printer. The commands are based on the Dynamark 3 Protocol document L027952, except appendices D and E.
Implementation of the Dynamark Protocol allows connection to 4 clients maximum at the same time in parallel. Using this feature can cause conflicting results, therefore it must be ensured that the different clients do not interfere against each other.

2.1. Interface configuration

Following configuration parameters will be implemented in the G-Series Service section:

Parameter Name	Description
Protocol type	This parameter defines which protocol is used for remote control. • None • Modbus • Dynamark 3 (Only one of these protocol can be used at the same time)
Port number	This parameter defines which port number is used for Dynamark 3 protocol. • 20000 Default for Dynamark 3
Echo	This parameter turns the echo for Dynamark 3 protocol on or off. Note: G-Series ignores the "Echo" configuration because it doesn't echo any received command. To manually test Dynamark commands, adjust local echo in the terminal program. • On Default for Dynamark 3 • Off
Minimum Buffer level	This parameter defines the buffer level which triggers transmission of MSG27 • 10 Default value There is no maximum buffer level, buffer size is organized dynamically. Only restriction is available memory in the printer.
Send group number	This parameter defines if G-Series software should append the print group number to an event message (MSGx, see chapter "**Fehler! Verweisquelle konnte nicht gefunden werden.**"). • On • Off Default for Dynamark 3

2.2. Tutorial

Following chapter explains the principle for communication with G-Series using Dynamark protocol.

2.2.1. Easy Interface connection

Ensure that the connection is working properly by entering any character which should reappear as echo from the printer (as long as the terminal program local echo is turned on).
Press ENTER (which should be configured as carriage-return plus linefeed) without previously entering any command and an error message (ERROR nnnn) will appear.
Try some remote control commands.
As a first step, load different print messages, the messages must be available in the controller:
For example create three different messages and store them using the names Label1.msg, Label2.msg and Label3.msg.
Enter in the remote terminal:

LOADPROJECT Label1.msg [ENTER]
OK
LOADPROJECT Label2.msg [ENTER]
OK
LOADPROJECT Label3.msg [ENTER]
OK

Different print messages can be opened from a remote peer. An error (ERROR 9) will be returned when trying to load a message that does not exist.

Another example:
Create a fourth message called "My Example.msg" and try to load it again:

LOADPROJECT My Example.msg [ENTER]
ERROR 2

This example did not work as expected because there is one parameter (the filename) in two words (the space character separates commands and parameters). In this case (or always if wanted), use the "character; the correct way is:

LOADPROJECT "My Example.msg" [ENTER]
OK

Next step can be to enable marking from such a message that has been successfully loaded into the controller – here we need to use the MARK-command. Try as following:

LOADPROJECT Label1.msg [ENTER]
OK
MARK START [ENTER]
OK

This example loads message Labevl1.msg and enables marking.
In most applications these two commands should be sufficient; sometimes there is a need to change the content of text or barcode objects after loading a message.

Create another message called Label5.msg and insert a text object called "Text 1" and a barcode object called "Barcode 1" into this message and save it. The following example shows loading of this message, changing from contents of barcode and text and enable marking.

LOADPROJECT Label5.msg [ENTER]
OK
SETTEXT "Text 1" "New Text" [ENTER]

OK
SETTEXT "Barcode 1" "New Barcode" [ENTER]
OK
MARK START [ENTER]
OK

The text should be printed as "New Text" and the human readable contents of the barcode object should be "New Barcode".
Text contents or other object attributes can be changed while marking is enabled.
Note that the OK result of a SETTEXT command or other message change commands only advises that the attribute has been set properly. There is no indication that the message contains the new data when a new print is triggered straight away.
If such information is required, refer to chapter messages.

2.2.2. Transactions

Previous examples showed how to load print messages, change text within those print messages and how to enable printing.
In reality, messages can be much larger with more text content changes than the examples given above. A lot of SETTEXT or SETDATA commands can be given before marking is enabled, but this practice leads to a mixture of communications containing half of the old and half of the new contents which can be not wanted. This problem can be overcome by using the Transaction function.
With this Transaction function a change can be prepared by sending all SETTEXT/ SETDATA commands and setting them "live" at a defined time with the EXECTRANS command.
As already explained this is also a command that returns OK after successful change of all attributes; it does not advise that the new data is compiled and already ready for printing.

Example for Transaction:
Load one of the messages from the previous examples and enable marking.

LOADPROJECT Label5.msg [ENTER]
OK
MARK START [ENTER]
OK

Start printing and carry out a few test prints, then enter the following commands, triggering prints whilst entering them:

BEGINTRANS [ENTER]
OK
SETTEXT "Text 1" "Domino" [ENTER]
OK
SETTEXT "Barcode 1" "Domino G-Series" [ENTER]
OK
EXECTRANS [ENTER]
OK

Note that the old print commands will be used even after the SETTEXT command. The new contents will not be printed before the EXECTRANS command is executed.

Remark 1:
OK indicates that the command was received properly and entered into the transaction list. OK does not indicate that the command and all its parameters are correct and can be executed without any failure with the command EXECTRANS.

Remark 2:
Commands which can be used inside a transaction are marked with "Transaction-Command = YES" in the following tables. "Transaction-Command = NO" means that the command will be executed immediately.

2.2.3. Message-IDs

It was mentioned in previous chapters that in case printing is enabled, commands like SETTEXT (changes the print message) with an OK result, do not indicate that this new content is ready for printing. After every change the printer software must update the complete print message first.
Therefore it can be useful to know when such a new content is really ready to print. In chapter 5 it is shown how message-ID 1 can be used to get this information.
The printer can send data over the open interface connection directly; a strict request-answer concept is not used. If a TCP/IP connection is used, it is recommended to use a second connection for these messages to prevent confusion on client software side, a message can come in while another command is send. Each message has to be enabled separately with the SETMSG command before. As default no Message Ids are send.
Example below enables a message, changes text content and wants to know when the controller is ready again with the new data:

```
SETMSG 1 1 [ENTER]
OK
LOADPROJECT Label5.msg [ENTER]
OK
MARK START [ENTER]
OK
BEGINTRANS [ENTER]
OK
  SETTEXT "Text 1" "Domino" [ENTER]
  OK
  SETTEXT "Barcode 1" "G-Series" [ENTER]
  OK
EXECTRANS [ENTER]
OK
MSG 1
```

The MSG 1 that is sent from the printer advises when new data is active and ready to print.

Refer to Chapter 5 for further information.

2.2.4. Print Groups

The G-Series printer can be configured to support up to four print groups. Each print group can print individual print messages. Therefore for some commands it is necessary to select the print group before the command is sent. These commands are specified with print group command "yes" in this specification. The print group is selected with the SELECTGROUP command or with the SETPARAM command and the number of print groups can be read by using the GETPARAM Command.

```
BEGINTRANS [ENTER]
OK
 SELECTGROUP 1 [ENTER]
 OK
 LOADPROJECT Label1.msg [ENTER]
 OK
 SETTEXT "Text 1" "Apples" [ENTER]
 OK
 SETTEXT "Barcode 1" "Granny Smith" [ENTER]
 OK
 MARK START [ENTER]
 OK
 SELECTGROUP 2 [ENTER]
 OK
 LOADPROJECT Label1.msg [ENTER]
 OK
 SETTEXT "Text 1" "Beer" [ENTER]
 OK
 SETTEXT "Barcode 1" "Samuel Adams" [ENTER]
 OK
 MARK START [ENTER]
 OK
EXECTRANS [ENTER]
OK
```

This example shows that you can load the same message to two different groups and that you can set the objects to print to a different content.

Remark:
The SELECTGROUP command is defined as "Transaction-Command=YES" and SETPARAM command as "Transaction-Command=NO". Example above would fail in case SETPARAM is used inside the transaction.

2.2.5. Escaping of "<"

To be compatible with other Domino products (for example Laser Printer) special care for the character "<" is necessary.
Reason is that Domino Laser Printer support multi line text objects. To start a new line the LF character is used internally, for transmission with the Dynamark Protocol the LF character is transformed into the sequence <10>.
If a "<" character is used inside a text, this character has to be escaped by a second "<" character for transmission. When the printer receives a "<" character the second "<" character that follows directly will be removed.
G-Series does not support multi line text objects, by escaping "<" as described it will be possible to extend the functionality without a need to modify the protocol.
If the current G-Series will receive the sequence <10> or just a single "<" it will take this as part of the text and will print this.

3. Dynamark Command Reference

The interface to the Dynamark 3 software is based on TCP/IP via Ethernet. The protocol is UTF-8-based.
Internally G-Series is capable to work inside the range of 16 bit Unicode characters.
Every command starts with its token which can be of different length (e.g. SETTEXT), followed by additional parameters, separated by using blanks; if needed, a parameter can be quoted using " and terminated by a Carriage-Return and a Linefeed character. The token is case sensitive.
Attention: It is important to understand that the interface protocol expects that the client application waits until the responses to commands are received - it is not allowed to send the next command while the printer software is still sending the response to the previous command!

3.1. 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 message will not be updated while the commands of the list are received. This saves execution 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).	
Parameter Description	
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
BEGINTRANS LOADPROJECT "BILBO.MSG" SETTEXT "Hometown" "Hobbingen" MARK START EXECTRANS This example loads the message BILBO.MSG, sets the contents of a text-object called "Hometown" to "Hobbingen" and enables marking.	
See also:	
EXECTRANS	

3.2. BUFFERCLEAR

Command	
BUFFERCLEAR	
Description	
This command clears all records from the remote queue (filled with the BUFFERDATA command).	
Parameter Description	
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
See also:	
BUFFERDATA, GETBUFFERSTATUS, MSG 27	

3.3. BUFFERDATA

Command	
BUFFERDATA <ID> <text 1> ... <text n>	
Description	
This command inserts a record into the remote queue to dynamically change objects of type CObjVarText for each print. For synchronization purposes, every record can be provided with a unique, numeric ID that will be returned by the messages 25 and 27. If not needed, the ID can be set to "-1" in order to automatically generate these IDs internally. If you use the automatic ID generation (ID=-1) there is a limitation of 9999 records that can be preloaded.	
Parameter Description	
ID	Unique identifier of the data record. Set this value to -1 if unique identifier is not necessary
Text	These parameters contain the coding data. The assignment of the parameters <text1> .... <textn> to the objects of type CObjVarText is done in the print message definition from the G-Series controller
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
BUFFERDATA 1 "record1 text1" "record1 text2" BUFFERDATA 2 "record2 text1" "record2 text2" This example writes two coding datasets to the Dynamark application.	
See also:	
BUFFERCLEAR, GETBUFFERSTATUS, MSG 25, MSG 27	

3.4. EXECTRANS

Command	
EXECTRANS	
Description	
Executes a transaction introduced with the BEGINTRANS command. Each transaction can only be executed once.	
Parameter Description	
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
BEGINTRANS LOADPROJECT "BILBO.MSG" SETTEXT "Hometown" "Hobbingen" MARK START EXECTRANS This example loads the message BILBO.MSG, sets the contents of a text-object called "Hometown" to "Hobbingen" and enables mark-mode.	
See also:	
BEGINTRANS	

3.5. GETBUFFERSTATUS

Command	
GETBUFFERSTATUS	
Description	
Use of this command obtains the number of data records in memory of the printer still to be printed.	
Parameter Description	
Result Description:	
Success:	RESULT GETBUFFERSTATUS <level>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
GETBUFFERSTATUS RESULT GETBUFFERSTATUS 3 The result of this request shows that the printer still has 3 prints in memory to be printed.	
See also:	
BUFFERDATA, BUFFERCLEAR, MSG 27	

3.6. GETCOUNT

Command	
GETCOUNT <Counter-ID>	
Description	
Reads the current formatted value of the specified counter. This may be a numerical value or a string if there is a conversion from numbers to characters active to print “145” as “ADE”.	
Parameter Description	
Counter-ID	Name of the user defined counter. G-Series contain 10 counters which are numbered from 1 to 10. Each counter can be named individually during configuration of the printer.
Result Description:	
Success:	RESULT GETCOUNT <Counter-ID> <Value>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
GETCOUNT batchn RESULT GETCOUNT batchn ADE	
See also:	

3.7. GETCOUNTERVALUE

Command	
GETCOUNTERVALUE <Counter-ID>	
Description	
Reads the current reference value of the specified counter. This is always the internal numerical value of the counter.	
Parameter Description	
Counter-ID	Name of the user defined counter. G-Series contain 10 counters which are numbered from 1 to 10. Each counter can be named individually during configuration of the printer.
Result Description:	
Success:	RESULT GETCOUNTERVALUE <Counter-ID> <Value>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
GETCOUNTERVALUE batchn RESULT GETCOUNTERVALUE batchn 145	
See also:	
SETCOUNTERVALUE	

3.8. GETCURRENTPROJECT

Command	
GETCURRENTPROJECT	
Description	
Determines the filename of the current message which is loaded for printing. The command returns an error if no message has been loaded. The syntax of the returned filename is the same as the one used by LOADPROJECT and SAVEPROJECT.	
Parameter Description	
Result Description:	
Success:	RESULT GETCURRENTPROJECT <filename>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
GETCURRENTPROJECT RESULT GETCURRENTPROJECT "BILBO.MSG" The result of this example shows that the current message is "BILBO.MSG".	
See also:	
LOADPROJECT, SAVEPROJECT	

3.9. GETDATE

Command	
GETDATE	
Description	
Gets time/date of the system Attention: Command does not allow getting the time of the G-Series printer with high precision. Precision depends on the reaction time of the client, transfer time of the network and reaction time of G-Series printer.	
Parameter Description	
Result Description:	
Success:	RESULT GETDATE <hour> <minute> <second> <year> <month> <day>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
GETDATE RESULT GETDATE 13 45 00 2004 09 02 The system time is the 2nd of September 2004 at 13:45:00 o'clock.	
See also:	
SETDATE	

3.10. GETMARKMODE

Command	
GETMARKMODE	
Description	
Determines the current state of the marking engine on the printer.	
Parameter Description	
Result Description:	
Success:	RESULT GETMARKMODE <mode> mode=0: marking is not enabled mode=1: marking is enabled
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
GETMARKMODE RESULT GETMARKMODE 1 This example tells the remote peer that the marking engine is currently enabled.	
See also:	
MARK	

3.11. GETOBJECTS

Command	
GETOBJECTS [Object-type]	
Description	
Determines all objects available as a list of strings. You may optionally add a type-identifier as parameter to filter the output to one object-type only. Attention: In G-Series only the object type CObjVarText can be modified with the SETTEXT command, all the other object types are read only.	
Parameter Description	
Object-type	Optional filter-string; can be • "CObjBarcode" for Barcodes (RO), • "CObjBitmap" for Bitmaps (Logos) (RO), • "CObjText" for fixed Texts (RO) • "CObjVarText" for variable Texts (RW) • "CObjCounter" for Counters (RO) • "CObjShiftCode for Shift Codes (RO) • "CObjDateTime" for Date and Time (RO)
Result Description:	
Success:	RESULT GETOBJECTS "<Name>" "<Name>" ....
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
For the following two examples we assume to have a message loaded in the printer containing 2 barcodes called "Barcode 1" and "Barcode 2", 2 Texts called "Text 1" and "Text 2". In the first example we request the names of all barcode objects within the message: GETOBJECTS "CObjBarcode" RESULT GETOBJECTS "Barcode 1" "Barcode 2" In the second example we request the names of all objects within the message: GETOBJECTS RESULT GETOBJECTS "Barcode 1" "Text 1" "Text 2" "Barcode 2"	
See also:	

3.12. GETPARAM

Command	
GETPARAM <param-id> [additional-param-ids] [...]	
Description	
This command is used for getting setup and runtime parameters of the printer	
Parameter Description	
InkLevel n	Ink level of cartridge in head n [0.0 ml to 400.0 ml]
NumberOfHeads	Number of print heads [1 to 4]
SelectedGroup	Number of selected print group [1 to 4]
NumberOfGroups	Number of print groups [1 to 4]
Result Description:	
Success:	RESULT GETPARAM <param-id> [additional-param-id] <param-value> [additional-param-values] [...]
Fault:	ERROR <no>
Properties:	
Transaction-Command	no
Print Group Command	no
Implemented since version	Dynamark G-Series V1.0
Examples	
Example 1: Get Number of print heads GETPARAM NumberOfHeads RESULT GETPARAM "NumberOfHeads" "2" *Example21: Get InkLevel of cartridge in head 2* GETPARAM InkLevel 2 RESULT GETPARAM "InkLevel" "2" "34.6"	
See also:	
SETPARAM	

3.13. GETPARSEDTEXT

Command	
GETPARSEDTEXT <object>	
Description	
Requests the text contents of the specified object. Dynamic text-contents like counters, shift codes, etc. will be parsed and returned as current value. Line breaks will be substituted with "<10>". Please note that this command will not return an automatically calculated checksum on barcode objects.	
Parameter Description	
Object	Name of the object of which the parsed text shall be obtained
Result Description:	
Success:	RESULT GETPARSEDTEXT "<object>" "<text> "
Fault:	ERROR <no>
Properties:	
Transaction-Command	no
Print Group Command	yes
Implemented since version	Dynamark G-Series V1.0
Examples	
For the following example we assume to have a message loaded with a barcode object called "Barcode 1" which has been configured to use the content of object "Counter 1" and the content of the variable text object "Partno 1". The first example shows the unparsed contents of the object's text property, the second example shows the parsed text. GETTEXT "Barcode 1" RESULT GETTEXT "Barcode 1" "PART: #Partno 1# S/N: #Counter 1#" GETPARSEDTEXT "Barcode 1" RESULT GETPARSEDTEXT "Barcode 1" "PART: 00004711 S/N: 12345"	
See also:	
GETTEXT, SETTEXT	

3.14. GETPROJECTS

Command	
GETPROJECTS	
Description	
Returns all messages that are stored in the message store on the internal flash memory.	
Parameter Description	
Result Description:	
Success:	RESULT GETPROJECTS [fiilename1] [filename2] ...
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Nol
Implemented since version	Dynamark G-Series V1.0
Examples	
GETPROJECTS RESULT GETPROJECTS "test.msg" "myproject.msg"	
See also:	
SAVEPROJECT	

3.15. GETSTATUS

Command	
GETSTATUS [all-messages]	
Description	
Determines the current 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.	
Parameter Description	
all-messages	optional parameter; if =1 you will get the information on all active status messages, e.g. RESULT GETSTATUS <severity1> <status-id1> "<text1>" <severity2> <status-id2> "<text2>"
Result Description:	
Success:	RESULT GETSTATUS <severity> <status-id> "<text>" Severity: • 0=information • 1=warning • 2=temporary fault • 3=critical fault • 4=critical fault (needs to be reset by hardware) Status-ID: see operations manual Text: The text returned here is the message-text in the language that is currently selected on the user-interface.
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
GETSTATUS RESULT GETSTATUS 1 5011 "Cartridge 1: Ink Level Low" The result of this example shows that the system is in "warning" status with alarm no 5011 (English text: "Cartridge 1: Ink Level Low").	
See also:	
MSG 5	

3.16. GETTEXT

Command	
GETTEXT <Object>	
Description	
Requests the text contents of the specified object as it is defined in the print message. Dynamic text contents like counters, shift codes, etc. will not be parsed and returned with their token representation. Line breaks will be substituted with "<10>". For bitmaps the name of the file will be returned.	
Parameter Description	
object	Name of the text or barcode object of which the unparsed text shall be obtained
Result Description:	
Success:	RESULT GETTEXT "<Object>" "<Text>"
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
For the following example we assume to have a message loaded with a barcode object called "Barcode 1" which has been configured to use the content of object "Counter 1" and the content of the variable text object "Partno 1". The first example shows the unparsed contents of the object's text property, the second example shows the parsed text. GETTEXT "Barcode 1" RESULT GETTEXT "Barcode 1" "PART: #Partno 1# S/N: #Counter 1#" GETPARSEDTEXT "Barcode 1" RESULT GETPARSEDTEXT "Barcode 1" "PART: 00004711 S/N: 12345"	
See also:	
GETPARSEDTEXT, SETTEXT	

3.17. GETXML

Command	
GETXML	
Description	
This command returns the currently loaded print message as XML stream (containing no carriage-return and no line-feed as these are the end-identifiers). The resulting XML stream is the content of the message file, which is saved in the internal Flash memory. It is not the current live data held in the main memory. The user may call SAVEPROJECT if he changed the message properties, then call GETXML to get the saved state.	
Parameter Description	
Result Description:	
Success:	RESULT GETXML <xml-code>
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
GETXML RESULT GETXML <?xml version="1.0"?><G-Series-Project title="example.msg" version="0.1"><Line>[Definition]</Line><Line>Aufloesung(V1.0)=0;0;1;0;</Line><Line>Abmessung(V1.0)=1500;0;127;0;</Line><Line>Direction(V1.0)=0;0;0;0;0;0;</Line><Line>Margin(V1.0)=0;0;0;0;</Line><Line>Group(V1.0)=0;0;1;0;0;0;</Line><Line>Mode(V1.0)=0;0;0;0;</Line><Line>Speed(V1.1)=10;0;0;0;100;</Line><Line>Intelligent MP(V1.0)=0;0;</Line><Line>Element]</Line><Line>Name(V1.0)=Text;</Line><Line>Typ(V1.0)=2;1;</Line><Line>Ausrichtung(V1.0)=0;0;</Line><Line>Position(V1.0)=450;22;</Line><Line>Invers(V1.0)=0;</Line><Line>Font(V1.0)=arial.etf;22;0;</Line><Line>Chimney(V1.0)=0;</Line><Line>Text(V1.0)=www.domino-printing.com;</Line><Line>[Element]<Line></Line><Line>Name(V1.0)=text;</Line><Line>Typ(V1.0)=2;1;</Line><Line>Ausrichtung(V1.0)=0;0;</Line><Line>Position(V1.0)=0;0;</Line><Line>Invers(V1.0)=0;</Line><Line>Font(V1.0)=arial.etf;32;0;</Line><Line>Chimney(V1.0)=0;</Line><Line>Text(V1.0)=Domino;</Line></G-Series-Project>	
See also:	
SETXML	

3.18. LOADPROJECT

Command	
LOADPROJECT <filename>	
Description	
Loads the specified message for printing.	
Parameter Description	
filename	This parameter references the message that shall be loaded. This parameter can have two forms • *<filename>: just a filename including the extension ".msg"* • "" (two double-quotes without a blank in between) for closing the current message, the print message will be unloaded
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	Yes
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
LOADPROJECT "hobbingen.msg" This example loads the message "hobbingen.msg" from the internal flash memory of printer controller.	
See also:	
SAVEPROJECT, GETCURRENTPROJECT	

3.19. MARK

Command	
MARK <action>	
Description	
This command controls (means enables or disables) the marking mode of the printer.	
Parameter Description	
Action	Action=START: Enables marking mode (if not already enabled). Action=STOP: Disables marking mode (if not already disabled)
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	yes
Print Group Command	yes
Implemented since version	Dynamark G-Series V1.0
Examples	
MARK START enables the marking mode	
See also:	
GETMARKMODE	

3.20. RESETSYSTEM

Command	
RESETSYSTEM	
Description	
This command resets (quits) pending errors.	
Parameter Description	
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	no
Implemented since version	Dynamark G-Series V1.0
Examples	
RESETSYSTEM OK	
See also:	

3.21. SAVEPROJECT

Command	
SAVEPROJECT <filename>	
Description	
Saves the current message using the filename specified as parameter.	
Parameter Description	
Filename	The message currently loaded into for printing shall be saved under the specified name: This parameter can have two forms • *<filename>: just a* filename including the extension ".msg" • *live:* will store the information on the current live message to the internal storage device so it can be restored after power cut.
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	

SAVEPROJECT "beutlin.msg"

This example will save the current message with its current settings in the file named "beutlin.msg" in the internal flash memory. If the file already exists, it will be overwritten. If the file name differs from the actual file name of the currently loaded print message, the origin file name and the origin settings won't change. In this case you will just save the current settings in another file.

SAVEPROJECT live:

This example will save the currently loaded print message in the internal flash memory. Three cases have to be considered:

- If it was a message previously loaded from an existing file, this file will be overwritten with the current settings.
- If it was a newly created message via the graphical user interface, the print message will be stored in a file named "new.msg". "new.msg" will be overwritten with the current settings every time the command SAVEPROJECT with the "live:" specifier is executed.
- If it was a newly created message by the SETXML command, the project name given in the XML root node's "title" attribute is chosen as file name.

See also:
LOADPROJECT

3.22. SELECTGROUP

Command	
SELECTGROUP <Group>	
Description	
Selects the print groups for the following commands	
Parameter Description	
Group	Group number [1 to 4]
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	Yes
Print Group Command	no
Implemented since version	Dynamark G-Series V1.0
Examples	
Please refer to the tutorial	
See also:	

3.23. SETCOUNTERVALUE

Command	
SETCOUNTERVALUE <Counter-ID> <Value>	
Description	
Sets the internal reference for the specified counter to the specified value.	
Parameter Description	
Counter-ID	Name of the user defined counter. G-Series printer contain 10 counters which are numbered from 1 to 10, each counter can be named individually during configuration of the printer.
Value	This parameter references the new reference value of the counter.
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
The following command will set the reference of counter “batchn” to the absolute value 100. SETCOUNTERVALUE batchn 100 OK GETCOUNTERVALUE batchn RESULT GETCOUNTERVALUE batchn 100	
See also:	
GETCOUNTERVALUE	

3.24. SETDATE

Command	
SETDATE <hour> <minute> <second> [year] [month] [day]	
Description	
Sets time/date of the system; setting the date is optional and can be left out. Attention: This command does not allow setting the clock in the G-Series printer with high precision. Precision depends on reaction time of the client, transfer time in the network and reaction time of G-Series printer.	
Parameter Description	
Hour	Numeric value for current hours of the daytime (0..23)
Minute	Numeric value for current minutes of the daytime (0..59)
Second	Numeric value for current seconds of the daytime (0..59)
Year	Optional parameter setting the current year
Month	Optional numeric parameter setting the current month (1..12)
Day	Optional parameter setting the current day (1..31)
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	no
Print Group Command	no
Implemented since version	Dynamark G-Series V1.0
Examples	
SETDATE 13 45 00 2004 09 02 OK This example sets the system time to the 2nd of September 2004 at 13:45.00.	
See also:	
GETDATE	

3.25. SETMSG

Command	
SETMSG <MsgID> <Mode>	
Description	
Use of this command automatically makes the printer send messages **on several events via this interface.**	
Parameter Description	
MsgID	Refer to page 43
Mode	Mode=0 (default) Disable the sending a message for the given event. Mode=1 Enable the sending a message for the given event.
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
Please refer to the tutorial	
See also:	

3.26. SETPARAM

Command	
SETPARAM <param-id> [additional-param-ids] <param-value> [additional-param-values] [...]	
Description	
This command is used for setting setup and runtime parameters of the printer.	
Parameter Description	
SelectedGroup	Number of selected print group [1 to 4]
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	No
Implemented since version	Dynamark G-Series V1.0
Examples	
Example 1: Set the selected print group SETPARAM SelectedGroup "3" OK The G-Series printers may be configured to print with up to four print groups. These print groups are independent in nearly all aspects except ink level which adheres to the print head number 1 to 4 and the counters which adheres to counter numbers 1 to 10.	
See also:	
GETPARAM	

3.27. SETTEXT

Command	
SETTEXT <Object> <Text>	
Description	
Sets the text attribute of the specified variable text object (COBjVarText).	
Parameter Description	
Object	Name of the object
Text	Text to be displayed by the object
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	Yes
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0
Examples	
SETTEXT "Text 1" "Halmackenreuther" Sets the content of the object “Text 1” to "Halmackenreuther".	
See also:	

3.28. SETXML

Command	
SETXML <XML code..>	
Description	
Creates a new message using the XML code given as parameter and automatically loads the message for printing. The created message can be saved in the internal flash card by using SAVEPROJECT. The XML code has to be free of carriage-return/linefeeds. Otherwise, it is automatically the whole remainder of the line after 'SETXML', so you are not allowed to use quotes around the argument – only replace '\' with '\\' and newlines with '\n'. The XML parser on the G-Series does not fully conform to the XML 1.0 specification, although the version attribute in the XML declaration is checked for the sake of completeness: <u>XML syntax restrictions:</u> • Empty elements are not supported • Character references and entity references are not supported • External entities are not supported • CDATA sections are not supported • Conditional sections are not supported • Attribute types, attribute defaults and attribute value normalization are not supported • DTD is not supported <u>Application restrictions:</u> • The only allowed tags are: – "G-Series-Project" representing the XML root node – "Line" representing a message file entry • The title attribute's value must not be empty. The project name must have the file extension ".msg". If the command "SAVEPROJECT live:" is put in execution, the project name is chosen as file name. • Take care of upper and lower cases ("xml" tag and "version" attribute within the XML declaration, "title" attribute, "G-Series-Project" tag, "Line" tag and so on) • The encoding attribute in the XML declaration is optional. If given, its value is restricted to UTF-8 (encoding="UTF-8"). If omitted, the default value is UTF-8. • The project's version attribute including the version number is checked for existence, but in the current implementation the value of the version number is not checked, i.e. a project version is currently not supported.	
Parameter Description	
XML code	XML-sources of a message.
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	No
Print Group Command	Yes
Implemented since version	Dynamark G-Series V1.0

Examples
SETXML <?xml version="1.0"encoding "UTF-8"?><G-Series-Project title="example.msg" version="1.0">"><Line>[Definition]</Line><Line>Aufloesung(V1.0)=0;0;1;0;</Line><Line>Abmessung(V1.0)=1500;0;127;0;</Line><Line>Direction(V1.0)=0;0;0;0;0;0;</Line><Line>Margin(V1.0)=0;0;0;0;</Line><Line>Group(V1.0)=0;0;1;0;0;0;</Line><Line>Mode(V1.0)=0;0;0;0;</Line><Line>Speed(V1.1)=10;0;0;0;100;</Line><Line>Intelligent MP(V1.0)=0;0;</Line><Line>Element]</Line><Line>Name(V1.0)=Text;</Line><Line>Typ(V1.0)=2;1;</Line><Line>Ausrichtung(V1.0)=0;0;</Line><Line>Position(V1.0)=450;22;</Line><Line>Invers(V1.0)=0;</Line><Line>Font(V1.0)=arial.etf;22;0;</Line><Line>Chimney(V1.0)=0;</Line><Line>Text(V1.0)=www.domino-printing.com;</Line><Line>[Element]</Line><Line>Name(V1.0)=text;</Line><Line>Typ(V1.0)=2;1;</Line><Line>Ausrichtung(V1.0)=0;0;</Line><Line>Position(V1.0)=0;0;</Line><Line>Invers(V1.0)=0;</Line><Line>Font(V1.0)=arial.etf;32;0;</Line><Line>Chimney(V1.0)=0;</Line><Line>Text(V1.0)=Domino;</Line></G-Series-Project> OK
This example sends a message as XML source to the printer.
See also:
GETXML, SAVEPROJECT

3.29. TRIGGER

Command	
TRIGGER	
Description	
For performance reasons and for positioning precision we strongly recommend to use the digital hardware print go input (D-Top).	
Parameter Description	
Result Description:	
Success:	OK
Fault:	ERROR <no>
Properties:	
Transaction-Command	no
Print Group Command	yes
Implemented since version	Dynamark G-Series V1.0
Examples	
TRIGGER OK	
See also:	

4. Error Codes

Error-Code	Description
0	No error occurred
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	Transaction 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 message
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 as the specified object name already exists
22	Object creation failed (e.g. the specified object type is not supported)
23	Operation not allowed
25	No permissions to perform this action
26	Remote data buffering is not active
27	Cold start procedure failed
28	print message creation failed
29	Internal Error occurring while logging in the communication interface user.
32	Setting or getting network address is not successful.
33	Current command failed because printer is still printing. It can only work after the printer stops.
1000	*1) *1) = G-Series specific error codes will start with number 1000. This will make sure that any new error code from a Domino Laser is not in conflict to a G-Series error code.

5. Message IDs

MsgID	Message	Description of Message
1	MSG 1 [group]	Ready to print: Printer is turned on and printing is enabled. This message is also sent if print message has been updated because of external changes (e.g. SETTEXT).
5	MSG 5 <Severity> <Status-ID> <Text> [group]	Status Change with information on the new state: Severity: 0=information 1=warning 2=temporary fault 3=critical fault 4=critical fault (needs to be reset by hardware) Status-ID: id as defined in the status configuration Text: Status text in the selected translation
18	MSG 18 <filename> [group]	A message with <filename> has been loaded for printing.
25	MSG 25 <ID> [group]	This message is returned in case remote data coding is active and a print has been triggered. The ID that is returned is the ID (used with the BUFFERDATA command) of the record that just has been printed.
27	MSG 27 [group]	The remote data buffer level has reached the warning level as configured in the general section of the Interface Setup. See also BUFFERDATA command.

6. References

Dynamark 3 Communication protocol; L027952 Issue 4 September 2011