GFM

GFM Barcode Code39 Generator

User's And Programmer's Guide

Version 1

GFM BarCode Code 39 Generator User and Programmer's Guide

PUBLISHED BY

Graphic Prepress Solutions

PostScript, the PostScript logo, Display PostScript, Adobe, the Adobe logo, Adobe Illustrator, TransScript, Charta, and Sonata are trademarks registered in the U.S.A. and other countries. Macintosh and TrueType are registered trademark of Apple Computer, Inc. Intel and Pentium are registered trademark of Intel Corporation. IBM and OS/2 are registered trademark of International Business Machines Corporation. Lotus is a registered trademark of Lotus Development Corporation. CodeView, Microsoft, Microsoft Press, Microsoft QuickBasic, MS, MS-DOS, Quick C, XENIX, Visual Basic, Win 32, Win32s are trademarks and Visual C++ and Windows NT are registered trademarks of Microsoft Corporation. Other brand or product names are the trademarks or registered trademarks of their respective holders.

This publication and the information herein is furnished AS IS, is subject to change without notice and should not to be construed as a commitment by Företagsnamn. Företagsnamn assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind with respect to this publication, and expressly disclaim any and all warranties of merchantalibity, fitness for particular purposes and noninfringement of third party right.

200008-01 1998-02-12

CONTENTS

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND INSTALLATION

This document describes how to install, configure and use the GFM BarCode Code39 Generator module. It is both a user's guide and a programmer's guide.

Overview

The purpose of the module is to create barcode labels that should be assembled on a page in a graphical digital environment. For example can the module be used in combination with the **PageImposer** software from Graphic Prepress Solutions.

The module creates an EPS file containing the barcode label. The normal would be the final page name at any position on the page. The positioning is not done by the module, the GFM BarCode Code39 Generator generates the EPS file. The barcode is created according to the Code39 scheme, see *Appendix C Code 39 Basics* for more information.

Basic Function

The function of the module is very easy. You will send the text that should be in the barcode, the full name of the EPS file to create and send a 'generate barcode' command to the module. The module will then generate the EPS file containing the barcode and save the file in a specified directory. Any other application can then use the barcode label and position it at any position on the page.

Installation

The module is contained on one diskette. The installation is performed in the same way as for all modules, with the GFM Module Installer. The module will be installed in a directory named BarCode under the Modules directory.

Note! Before the installation, read the README.TXT file on the diskette since it may contain information that is more up to date than this document.

To install the module follow these instructions

1. Insert the GFM BarCode Creator module diskette in the drive.
2. Start the GFM Module Installer.
3. Select New Module and click on Install.
4. Select A:\ as path and click on OK

Follow the instructions given by the installer program.

For more information about how to use and install modules with the GFM Module Installer, see the document *Graphics Flow Mark User's Guide*.

After the installation will the module be available as any other module. The following section will describe how to configure the module

SECTION 2

CONFIGURING THE BARCODE CODE39 GENERATOR

This section will discuss how to configure the GFM BarCode Code39 Generator module. If you don't configure the module will it set all options to default values as described below.

Note! The easiest way to configure the module is to use the GFM Module Configuration Tool that is a standard software delivered with the GFM engine. When using this tool you will do the configuration via a menu. How to do this see the part *Configure Menu* later in this section.

When configuring the module you will tell the GFM BarCode Code39 Generator module a file name as the **SourceFileName** property and then set the **Action** property to 5. This will cause the module to read the specified file and parse the file. It will then set the options you have selected. If the file specified can't be found will there occur an error, error number 1, and the module will be left in error mode.

The Configuration File

The configuration file has a GFM syntax, as the example below:

```
(GFMBarCodeCreatorModule)
/ViewInformation 1 def
/NarrowWidth 1.6 def
/Height 20 def
/UseCheckCharacter 0 def
/Verbosity 0 def
```

The following will discuss each line in the configuration file and what options that are available for each setting. Note that all settings and options are case sensitive as in an ordinary GFM scripts.

(GFMBarCodeCreatorModule)

The first line is an identifier and must be the first line. If the first line in the configuration file isn't (GFMBarCodeCreatorModule) will an error occur, error number 2, and the configuration file will be rejected and the module left in error mode.

/ViewInformation *value* def

This line tells the module if the information menu shall be visible or not. The value has the following meaning:

Value	Meaning
0	The information menu will not be visible.
1	The information menu will be visible.

Default value is 0. If the value isn't 0 or 1 will an error occur, error number 3, and the module will be left in error mode.

/NarrowWidth *value* def

This defines the width in mm for narrow lines and narrow gaps. The wide lines and gaps will always be in the ratio 3:1, or 3 times the

narrow width. For more information about narrow lines and narrow gaps, see *Appendix C Code 39 Basics*.

Default value is 0.6, i.e. narrow lines and narrow gaps will be 0.6 mm. If the value isn't a number will an error occur, error number 3, and the module will be left in error mode.

/Height *value* def

This defines the height of the barcode in mm. For more information about the height of the barcode, see*Appendix C Code 39 Basics*.

Default value is 20, i.e. the height of the barcode will be 20 mm. If the value isn't a number will an error occur, error number 3, and the module will be left in error mode.

/UseCheckCharacter *value* def

Tells the module if the check character shall be used/inserted or not. For more information about the check character see*Appendix C Code39 Basics*. The value has the following meaning:

Value	Meaning
0	The check character shall not be used.
1	The check character shall be used.

Default value is 0, i.e. the check character shall not be used. If the value isn't 0 or 1 will an error occur, error number 3, and the module will be left in error mode.

/Verbosity *value* def

Tells the module the verbosity level. The verbosity level is the amount of messages the module shall generate. The use of the verbosity setting can be useful when doing fault finding, for example if a file will not be created correctly. The messages sent out from the module is written down to a file in the GFM main directory, usually C:\GFM, and is called BarCode.TXT. The value has the following meaning:

Value	Meaning
0	No messages.
1	Report each character as it is created. Also reports each step in the barcode creation process.
2	The same as setting 1, and the EPS code will also be written to the file.

Default value is 0, i.e. no messages will be written to the file. If the value isn't 0, 1 or 2 will an error occur, error number 3, and the module will be left in error mode.

An Example

The following example shows how to load a configuration file called C:\GFM\MODULES\BarCode\Config1.gfm.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BarCode\Config1.gfm)
!SourceFileName
5 !Action
```

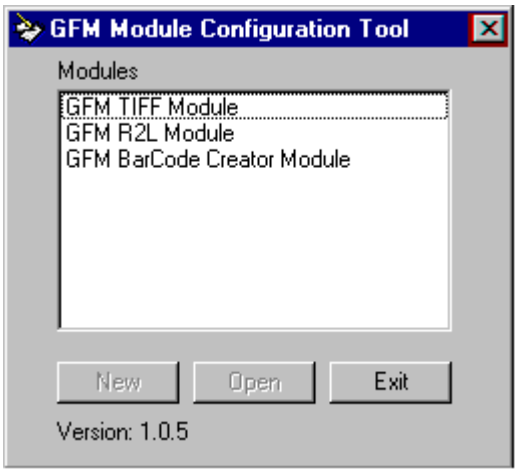
Configure Menu

There is a configure menu avaible to make the configuration easier. To access the configuration menu you specify the configuration file as the **SourceFileName** property and set the **Action** property to 1. To generate a new configuration you shall set the **SourceFileName** property to 'New File', the **DestinationFileName** property to the new file name and finally the **Action** property to 1.

Use The GFM Configuration Tool

The easiest way to create the configuration files is to use the GFM Configuration Tool that is supplied as a standard software with the GFM engine. How to use the GFM Configuration Tool is explained in the GFM User's Guide. There will be a short discussion how to use this tool with the GFM BarCode Creator module here:

1. Start the GFM Configuration Tool. This is the file CONFIG.EXE in the GFM Main directory.



2. Select the GFM BarCode Creator Module in the list box.

3. To create a new configuration, click on the **New** button and give the new configuration a name in the standard menu that appears. The following menu will appear:

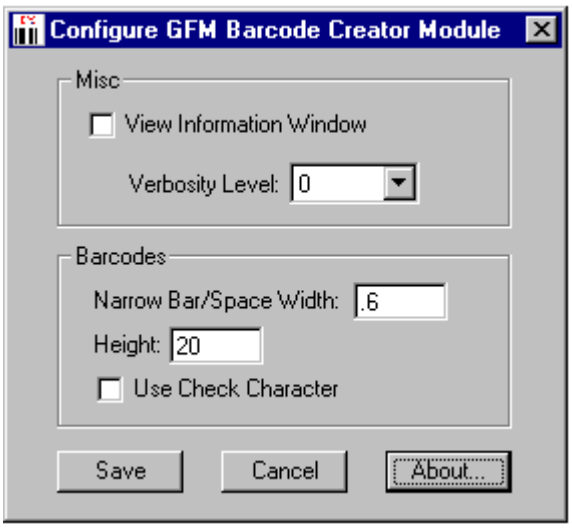


Figure *The GFM BarCode Code39 Generator Configuration Menu*

4. To edit an existing configuration click the **Open** button and select the file in the standard menu that appears.

5. Do your settings and click the **Save** button to save your settings or the **Cancel** button to abandon the configuration.

When you later writes your GFM script shall you load the configuration as described earlier in this section.

SECTION 3

WRITING GFM SCRIPTS FOR THE GFM BARCODE CREATOR

This section will discuss how write GFM scripts for the GFM BarCode Code39 Generator module and explain the properties and methods avaible.

When writing a script for the GFM BarCode Code39 Generator module you must be familiar with the GFM script language and the GFM environment. To have more information about the GFM script language see the documents*GFM Programmer's Guide* and *GFM Language Reference Guide.*

For details about all properties and methods see*Appendix A Programmer's Reference* in this document.

Principles

Writing scripts for the GFM BarCode Code39 Generator module is not very difficult. First you will connect to the module. Optionally will you then load a configuration file. The next step is to tell the module the original file name by setting the**SourceFileName**property, and

the resulting file as the **DestinationFileName** and finally set the **Action** property to a value telling the module what to do with the file.

Writing A Script

This part will go through the complete process how to write a script to the GFM BarCode Code39 Generator module.

Connect To The Module

You will connect to the GFM BarCode Code39 Generator module as all with all other modules with by pushing the connect name to the stack and run the **connectmodule** command as shown below.

```
(GFMBarCode.code39) connectmodule
```

Load Configuration File

When the module starts will it be set to default values as described in section 2. If you wants to change any of these settings you must load a configuration file. This is done by setting the configuration file name as **SourceFileName** and then set the **Action** property to 5. To have information about how to create configuration files, see section 2 in this document. The following example load a configuration file named C:\GFM\MODULES\BARCODE\MyConfig.GFM

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
```

Setting SourceFileName

When setting the text that should be the barcode label as the **SourceFileName** property it is important to understand that the GFM BarCode Code39 Generator module expects the text as a string. The following example sets up the text to be EX03.

```
(GFMBarCode.code39) connectmodule
```

```
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
(EX03) !SourceFileName
```

Setting DestinationFileName

When setting the EPS file name as the **DestinationFileName** property the directory where to save the new file must exist. The following example sets up the destination file name to be C:\BARCODES\EX03.EPS.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
(EX03) !SourceFileName
(C:\BARCODES\EX03.EPS) !DestinationFileName
```

Setting The Action Property

With the **Action** property will you now tell the module to create the EPS file. This is done by setting the **Action** property to 2.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
(EX03) !SourceFileName
(C:\BARCODES\EX03.EPS) !DestinationFileName
2 !Action
```

Disconnecting From The Module

To disconnect from the module you uses the **disconnectmodule** module as with all GFM modules. The following example disconnects from the GFM BarCode Code39 Generator module.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
(EX03) !SourceFileName
(C:\BARCODES\EX03.EPS) !DestinationFileName
2 !Action
```

```
disconnectmodule
```

Tidy Up After Script

The normal tidying up after a script is to delete the original file name and close all files. Since we don't have a original file in this case we just closes all files. This example closes all files.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action
(EX03) !SourceFileName
(C:\BARCODES\EX03.EPS) !DestinationFileName
2 !Action
disconnectmodule
closeallfiles
```

Writing A Script With GlobalStrings

The example above assumes that the script knows the file name by an absolute path and that the text that should be in the barcode label is fixed. It's not likely that you will write a script like this since the destination file name will probably be sent to the script via the GlobalStrings and also the text. The following example shows how to write a script where the destination directory is sent as GlobalString(1) and the destination file name is sent as GlobalString(2). The text that should be in the barcode label will be in the GlobalString(3).

```
% Definitions

/sDestinationFileName 2 getglobalstring def
/sDestinationDirectory 1 getglobalstring def
/sDestinationPath sDestinationDirectory (\)
mergestrings
sDestinationFileName mergestrings def
/sBarCodeText 3 getglobalstring def

% End definitions

% Load config file
(GFMBarCode.code39) connectmodule
```

```
(C:\GFM\MODULES\BARCODE\MyConfig.GFM)
!SourceFileName
5 !Action


sBarCodeText !SourceFileName
sDestinationPath !DestinationFileName
2 !Action
disconnectmodule

% Tidy up after script
closeallfiles
```

Now, this was just an example that you can use to get an overview how to write GFM scripts to the GFM BarCode Code39 Generator module.

APPENDIX A

PROGRAMMER'S REFERENCE

Module name: GFM BarCode Code39 Generator

Connect name: GFMBarCode.code39

ModuleType: 5

Allowed to configure: Yes

The next part of this appendix explains each property and method available in the GFM BarCode Code39 Generator module in detail. Note that all properties and methods are case sensitive.

Action property

Syntax *value* **!Action**

Description Sets the Action to be done by the GFM BarCode Code39 Generator module.

Remarks The type of action to be done is determined by the *value* according to the following table:

Value	Meaning
1	**Show the configure menu.** If SourceFileName is set to 'New File' will you be able to create a new configuration file. In that case you must set DestinationFileName to the new file name. If you wants to edit an existing configuration file the SourceFileName must have the path to the configuration file.
2	**Create EPS file.** When Action is set to 2 will the module create the barcode as the text in the SourceFileName property. The file name to be created must be in the DestinationFileName property.
5	**Load configuration file**. When Action property is set to 5 will the module expect a valid configuration file in the SourceFileName property. The module will load the configuration file and configure the module accordingly.

See also !SourceFileName, !DestinationFileName

Example This example connects to the GFM BarCodeCode39 Generator module, loads a configuration file named C:\GFM\MODULES\BarCode\CONF1.GFM and then create

the barcode EX04 and save the file as C:\EPSPAGES\BARCODE1.EPS.

```
(GFMBarCode.code39) connectmodule
(C:\GFM\MODULES\BARCODE\CONF1.GFM)
!SourceFileName
5 !Action
(PAGE1) !SourceFileName
(C:\EPSPAGES\BARCODE1.EPS)
!DestinationFileName
2 !Action
```

Errors no module connected, Stack underflow, Typecheck

>AllowConfigure property

Syntax >**AllowConfigure**

Description Will return the value 1 since the module can be configured.

See also connectmodule, !AllowConfigure, !Direction

Example This example connects to the GFM BarCode Code39 Generator module and checks if the module can be configured.

```
(GFMBarCode.code39) connectmodule
>AllowConfigure
```

Errors Stack underflow, Typecheck

!DestinationFileName property

Syntax *filename* **!DestinationFileName**

Description Sets the complete file name of the EPS file to create. With complete means the absolute path and the file name.

Remarks The meaning of the **!DestinationFileName** depends on **!Action** property of the module.

See also !Action, !SourceFileName

Example This example connects to the GFM BarCodeCode39 Generator module and writes the string value of the text that should be in the generated barcode label and sets the EPS file name to C:\LABELS\LABEL1.EPS.

```
(GFMBarCode.code39) connectmodule
(PAGE1) !SourceFileName
(C:\LABELS\LABEL1.EPS)!DestinationFileName
2 !Action
```

Errors stack underflow, typecheck

ErrorCond property

Syntax **ErrorCond**

Description Used by the GFM engine to monitor if there has occurred an error in the module. The ErrorCond will return a value indicating if there has occurred an error. If no error has occurred will ErrorCond contain the value 0. For a list over errors see Appendix C in this document.

Note! You shall never use this property in a GFM script.

See also ErrorMessage

ErrorMessage property

Syntax **ErrorMessage**

Description Used by the GFM engine to monitor if there has occurred an error in the module. The ErrorMessage will return a string

containing the text for the error. For a list over errors see Appendix C in this document.

Note! You shall never use this property in a GFM script.

See also ErrorCond

Init method

Syntax **Init**

Description Initializes the module. This is done by the GFM engine each time a script is sent to the module.

Note! You shall never use this method in a GFM script

>ModuleType property

Syntax >**ModuleType**

Description Returns the value 5 since this the GFM BarCodeCode39 Generator module is a stand-alone module.

Example This example connects to the GFM BarCodeCode39 Generator module and reads the module type.

```
(GFMBarCode.code39) connectmodule
>ModuleType
```

Errors Stack underflow, Typecheck

!SourceFileName

Syntax *text* **!SourceFileName**

Description Sets the modules source file name that usually is the text that should be in the barcode label.

Remarks The meaning of the **!SourceFileName** depends on setting of the **!Action** property.

See also !Action, !DestinationFileName, >ModuleType,

Example This example connects to the GFM BarCodeCode39 Generator module and writes the string value of the text that should be in the generated barcode label and sets the EPS file name to C:\LABELS\LABEL1.EPS.

```
(GFMBarCode.code39) connectmodule
(PAGE1) !SourceFileName
(C:\LABELS\LABEL1.EPS)!DestinationFileName
2 !Action
```

Errors Stack underflow, Typecheck

>Version property

Syntax **>Version**

Description Returns a string value telling the current version of the GFM BarCode Code39 Generator module.

Remarks The value returned will be on top of the stack after the command has been sent to the module.

Example This example connects to the GFM BarCode Code39 Generator module and checks the version of the module.

```
(GFMBarCode.code39) connectmodule
>Version
```

Errors no connected module

APPENDIX B

ERROR MESSAGES

This appendix describes all error messages that the module can generate. The normal error convention is used: error numbers in the range from 0-9 are reserved for the system. Errors in he range from 10-19 are barcode related errors and error numbers in the range 20-29 are EPS file related.

Each error begins with either BARCODE or EPS. All errors beginning with BARCODE are generated when creating the barcode. All errors starting with EPS are generated when writing the EPS file.

Error number	Error message	Remarks
0	No error	
1	Configuration file not found	
2	Not a valid configuration file	
3	Not a valid setting in the configuration file	
4	CRITICAL: Could not find the	

	GFM engine registry	
10	BARCODE: Not a valid character in the Code 39 character set	
11	BARCODE: Narrow bar/space is 0	
12	BARCODE: Label height is 0	
13	BARCODE: Label width is 0	
14	BARCODE: No text to display in the barcode label	
15	BARCODE: Margin is 0	
16	BARCODE: Unkown definition in barcode	
17	BARCODE: Could not create the check character (out of range)	
18	BARCODE: Could not create the check character (symbol not found)	
20	EPS: Could not create EPS file. Path not found	

APPENDIX C

CODE 39 BASICS

This appendix describes the basic construction of the Code 39 scheme. It is not ment to be a complete specification but only a brief description about how to create and decode a code 39 bar code.

Code Construction

Code 39 is an alphanumeric bar code that can encode decimal numbers, the upper case alphabet, and the following special symbols:

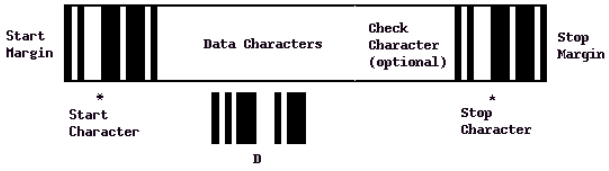
_ . * $ / % +

Code 39 characters are constructed using nine elements, five bars and four spaces. Of these nine elements, two of the bars and one of the spaces are wider than the rest. Wide elements represent binary ones (1), and narrow elements represent binary zero (0). The character set table below shows each of the available characters with their corresponding check character values.

To enable a decoder to distinguish between the wide and narrow elements a minimum wide to narrow ratio is needed. Depending upon which resolution has been used for the printing of the bar code, the width of the wide element should be at least two times greater than the narrow element. A ratio of three to one is better. All elemens of the same type should be printed at the same size. The width of a narrow bar should be the same as a narrow space.

Code 39 is a discrete bar code, with a space between characters which contains no information. The width of this intercharacter gap should be approximately equal to the narrow element width. The structure of Code 39 makes it self checking, but there is an optional message check character.

Code Structure



Character Set

ASCII Character	Binary Word	Bars	Spaces	Check Character Value
0	000110100	00110	0100	0
1	100100001	10001	0100	1
2	001100001	01001	0100	2
3	101100000	11000	0100	3
4	000110001	00101	0100	4
5	100110000	10100	0100	5
6	001110000	01100	0100	6
7	000100101	00011	0100	7
8	100100100	10010	0100	8
9	001100100	01010	0100	9
A	100001001	10001	0010	10
B	001001001	01001	0010	11
C	101001000	11000	0010	12
D	000011001	00101	0010	13
E	100011000	10100	0010	14
F	001011000	01100	0010	15
G	000001101	00011	0010	16
H	100001100	10010	0010	17
I	001001100	01010	0010	18
J	000011100	00110	0010	19
K	100000011	10001	0001	20
L	001000011	01001	0001	21
M	101000010	11000	0001	22
N	000010011	00101	0001	23
O	100010010	10100	0001	24
P	001010010	01100	0001	25
Q	000000111	00011	0001	26
R	100000110	10010	0001	27
S	001000110	01010	0001	28
T	000010110	00110	0001	29
U	110000001	10001	1000	30
V	011000001	01001	1000	31
W	111000000	11000	1000	32
X	010010001	00101	1000	33
Y	110010000	10100	1000	34
Z	011010000	01100	1000	35

-	010000101	00011	1000	36
.	110000100	10010	1000	37
SPACE	011000100	01010	1000	38
*	010010100	00110	1000	-
$	010101000	00000	1110	39
/	010100010	00000	1101	40
+	010001010	00000	1011	41
%	000101010	00000	0111	42

Check Character

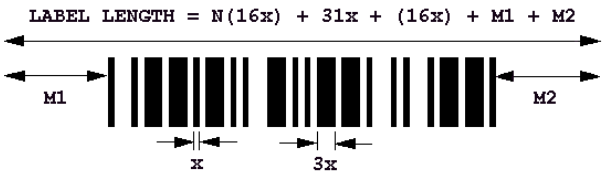
The Code 39 check character is a modulus 43 sum of all of the message character values and is printed as the last character in the message. The check character values are given from the table below. The check character is computed by adding up all of the values, dividing by 43, and using the remainder as the value of the check character.

Example

```
Message:        CODE 39
Characters:     C   O   D   E       3  9
Value:          23  24  13  14  38  3  9
Sum of values: 113
                113 / 43 = 2 remainder 27
                      27 = R check character
```

Label Length

The length of a printed Code 39 label can be determined from the following formula, once the ‘x’ dimension is known. The ‘x’ dimension is the width of the narrowest bar or space in the bar code, and all bars and spaces are a multiple of this number. The following formula assumes a 3:1 wide to narrow ratio.



Parameters:

M1, M2	Margins. These should be 6 mm or 10 times the narrow element width, whichever is greatest.
N	The number of data characters
16x	Width of the data characters, including the intercharacter gap (assumes a 3:1 wide to narrow ratio).
31x	Width of the start and stop characters. Includes the intercharacter gap between the start character and the first data character.
(16x)	Width of the optional check character.