

Volume

1

MOROVIA CORPORATION

Department of Research & Development

PDF417 Fontware Developer's Guide

MOROVIA CORPORATION

PDF417 Fontware Developer's Guide

© Morovia Corporation
88 Corporate Dr., #105
Toronto, ON M1H3G6
Phone 416.296.9605 • Fax 416.296.9178

Table of Contents

Introduction.....	1
Major features of PDF417 symbology	1
Character Set – All 128 ASCII characters, All 128 extended ASCII characters, 8-bit binary data;	1
Symbol Size – 3 to 90 rows, 90 to 583X in width	1
Bidirectional Decoding – Yes	1
Error Correction Level – 0 (no error correction) to 8 (the maximum error correction level).....	1
Additional Options – Macro PDF417, Truncated PDF417, Global Label Identifier(GLI)	1
Morovia PDF417 Fontware	1
Programming PDF417.....	2
How PDF417 Fontware works?.....	2
Security Level	3
Truncated Symbol.....	3
Number Of Columns	3
Number Of Rows	4
Special Characters.....	4
Using PDF417 Encoder with Visual Basic.....	4
Using PDF417 Encoder with Visual C++.....	5
Index	9

Introduction

PDF417 is a multi-row, variable-length symbology with high data capacity and error-correction capability. PDF417 has some unique features which makes it the widely used 2D symbology. A PDF417 symbol can be read by linear scanners, laser scanners or two-dimensional scanners. PDF417 is capable of encoding more than 1100 bytes, 1800 text characters or 2710 digits. Large data files can be encoded into a series of linked PDF417 symbols using a standard methodology referred to as Macro PDF417.

Major features of PDF417 symbology

Character Set – All 128 ASCII characters, All 128 extended ASCII characters, 8-bit binary data;

Symbol Size – 3 to 90 rows, 90 to 583X in width

Bidirectional Decoding – Yes

Error Correction Level – 0 (no error correction) to 8 (the maximum error correction level)

Additional Options – Macro PDF417, Truncated PDF417, Global Label Identifier(GLI)

Morovia PDF417 Fontware

Morovia PDF417 Fontware provides a font-based solution of printing PDF417 symbols from other applications. There are many advantages of employing a font-based solution. Fonts contain drawing commands in stead of the information of the pixels, thus they are device-independent. The artwork you created in one platform is available to other platforms, as long as you manage to have the fonts available. The latter can be done either by installing the fonts in the target machine or using font

embedding technology. Many applications, such as Adobe Acrobat and Crystal Reports, do not support ActiveX thus the fonts are the only solution.

Although it is a font-based solution, the character set of the font is relatively simple (it contains 16 characters, 0-9 and A-F). The comprehensive logic is hidden in the encoder. On Windows platforms, the encoder is provided as an Active control which does not have any GUI interfaces. In the future we will make the encoder available in binary forms in Macintosh and Unix platforms. We also make the source code available at a premium. The support of platforms outside Windows is not a topic in this guide. If you need to use the font products in other platforms please contact Morovia Sales department at sales@morovia.com.

Morovia PDF417 Fontware consists of the following components:

- Morovia PDF417 Fonts (5 typefaces: MRV PDF417XS, MRV PDF417S, MRV PDF417M, MRV PDF417T, MRV PDF417XT)
- Morovia PDF417 Encoder ActiveX
- Morovia PDF417 Encoder GUI
- Morovia 2D UFL for Crystal Reports
- Samples (Microsoft Access, Mail Merge and Crystal Reports)

Programming PDF417

As mentioned in the chapter 1, the encoding logic resides in the PDF417 Encoder, not the font. The font is relatively simple and only contains a couple of patterns. The PDF417 Encoder is packaged in an ActiveX in Windows platforms. A Java-based encoder is scheduled to be released at the end of July, 2004.

How PDF417 Fontware works?

There are five fonts in the Morovia PDF417 font family. The typeface begins with MRV which is the vendor ID for Morovia Corporation, Font department. Each font has a different height/width ratio ranging from 3 to 7:

Typeface	Cell Aspect Ratio
MRV PDF417XS	3
MRV PDF417S	4
MRV PDF417M	5
MRV PDF417TS	6
MRV PDF417XT	7

To generate a PDF417 symbol, retrieve the encoding results (called barcode string) from the PDF417 encoder (either from GUI or ActiveX) and format them with the appropriate PDF417 font:

```
00AB95CFCB0820CA1116F845CC563E02EB7
00AB82BE4D1DE3E9D0B7B8EC4C15CE02EB7
00AB9583DDF4253811B82A310C0A5E02EB7
00AB85A1CEB3E44F76E67BC8E5410602EB7
00AB9427C7D963B131BE68A77CA3DE02EB7
00AB850DC0D3AE1BD31B890B9C507602EB7
00AB9638421DA508369CF9E75D631E02EB7
00AB816C5051E3ECB3B1789C6C2D7E02EB7
00ABACC1D9F62CE4370F6B390C163602EB7
00ABAE31CCB7EB3CF26EFAF39D73E602EB7
00AB963F4BF1617DD1FB2A0A3CB1DE02EB7
00ABAEF0D827A8B9F67D1A620C171602EB7
00AB8BE8C90C23CD9116F845CD78E202EB7
00AB85C9CD1DE3E9B7A1E85D7C2EEE02EB7
00AB97D05C3DAC12F6118905DC6A0602EB7
00AB8D21D9CFAE9CF672F9EDE4D79E02EB7
00ABAE044288EEC370B7BA604D70FA02EB7
00ABB4604A05EF64171DCBB61C683602EB7
00AB8979C475EF49F0849842BC4B9E02EB7
00AB85F748342263B0B93857DD782602EB7
00AB8DA04853E052F32C3BC12DB78602EB7
```

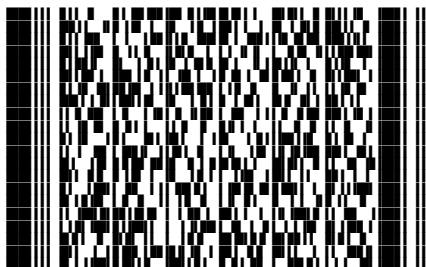
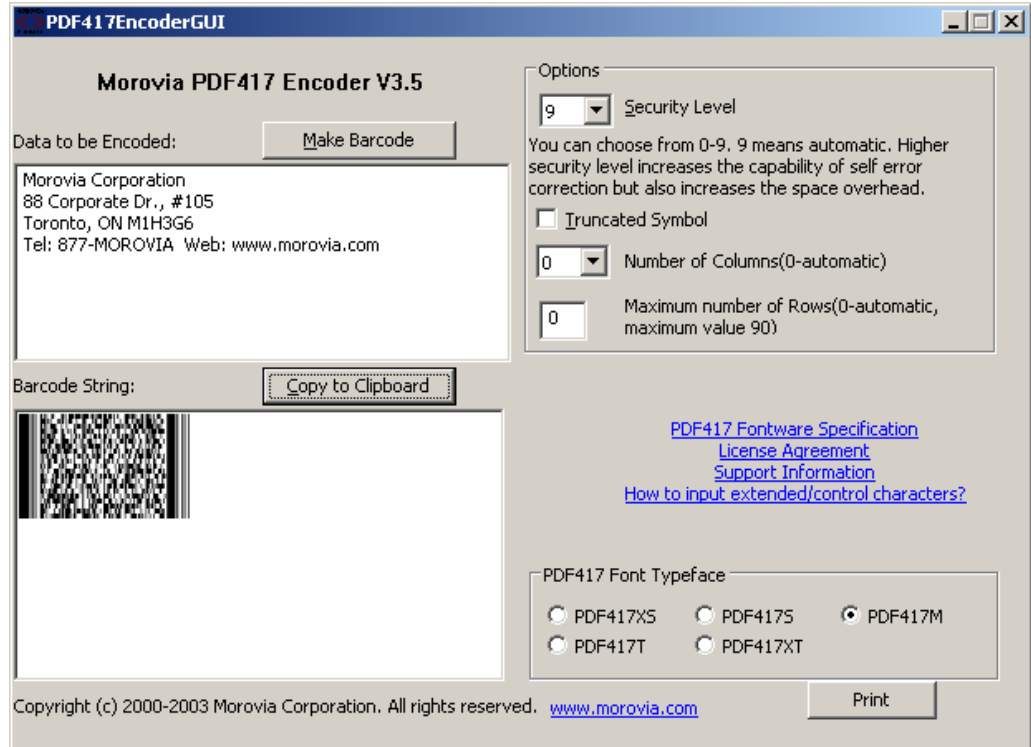


Figure 1 PDF417 Generated using PDF417 Font. The left side is the barcode string which becomes a PDF417 barcode after formatted with PDF417 Font.

To change the symbol size, you just need to change the point size used. We recommend you use a point size ranged between 3 and 20. (value 6 is preferred).

PDF417 Encoder (GUI)

The GUI version of the PDF417 encoder can be found under the shortcut Morovia|PDF417 Fontware.



Press copy you copies both font information and barcode string into the clipboard. To receive the barcode string in text form, paste into a text-only program such as Notepad.

There is four parameters you can adjust: security level, truncated symbol, number of columns and number of rows. Each parameter impacts the final barcode string.

Security Level

You can select a value from 0 to 8 for the security level used during encoding process. The higher the value is, the more secure the symbol. However, it also dramatically increases the space overhead. If you choose 9, the program will pick the best security level for you, based on the data encoded.

Truncated Symbol

A PDF417 symbol can be truncated to save the space. Under this option, the right stop pattern is reduced to a single bar.

Number Of Columns

A PDF417 symbol can have 1 to 30 columns in the data area. If you select 0 for this parameter, the program automatically chooses the number of columns.

Number Of Rows

A PDF417 symbol can have 3 to 90 rows in the data area. If you select 0 for this parameter, the program automatically chooses the number of rows.

Special Characters

PDF417 can also encode characters with values greater than 127. It can not encode double-byte characters. It can also encode control characters. To put a special character into the input, use the following form:

`\ddd`

Where ddd is the three-digit ASCII value for the character. For example, to encode special character carriage return (`\r`), just type `\013` instead. To encode backslash sign (`\`), type two backslash signs.

PDF417 Encoder (ActiveX)

The engine of the PDF417 encoding lies in this ActiveX dll, PDF417FontEncoder.dll, which is usually located under program files\morovia\PDF417fontware folder. It is a self-registered ActiveX component. To register the component, just runt the command under DOS prompt:

```
Regsvr32 PDF417FontEncoder.dll
```

This DLL also exposes Windows DLL interface so you can treat it as a normal Windows DLL if you are more familiar with programming with DLL interface.

Features of Morovia PDF417 Encoder ActiveX:

DLL Name: PDF417FontEncoder.dll

Prog ID: Morovia.PDF417FontEncoder

Properties: PDFMaxCols, PDFMaxRows, PDFSecurityLevel, PDFTruncatedSymbol, LineFeedString

Methods: Encode

Using PDF417 Encoder with Visual Basic

```
Dim encoder As Object

Dim output As String

Set encoder = CreateObject("Morovia.PDF417FontEncoder")

output = encoder.Encode(DataToEncode)
```

Using PDF417 Encoder with Visual C++

```

CComQIPtr<IDispatch> disp;

hr = disp.CoCreateInstance(L"Morovia.PDF417FontEncoder");

LPOLESTR szNameEncode = L"Encode";

LPOLESTR szNameLineFeedString = L"LineFeedString";

DISPID dispid_targetsize, dispid_encode, dispid_linefeedstring;

hr = disp->GetIDsOfNames(IID_NULL,      &szNameEncode,      1,
                        LOCALE_SYSTEM_DEFAULT,      &dispid_encode);

hr = disp->GetIDsOfNames(IID_NULL,      &szNameLineFeedString,      1,
                        LOCALE_SYSTEM_DEFAULT,      &dispid_linefeedstring);

_com_dispatch_propput( disp, dispid_targetsize, VT_I4, target_size_id);

_com_dispatch_propput( disp, dispid_linefeedstring, VT_BSTR, strLineFeed);

CComBSTR bstrResult;

_com_dispatch_method( disp, dispid_encode, DISPATCH_METHOD,
                    VT_BSTR, static_cast<void*>(&bstrResult),      L"\x0008",
strDateToEncode);

```

Morovia 2D UFL for Crystal Reports

To facilitate customer generating PDF417 symbols in Crystal Reports forms, we provide the UFL (User Function Library). Note that you must use the one for 2D symbologies, not the one for linear symbologies. This DLL is normally located at system32 folder under Windows directory. The name of the DLL is `crufMrv2DSurrogate.dll`. All 2D symbology font products share the same DLL.

Note: the internal implementation of the UFL functions use the encoder ActiveX DLL so when you distribute the reports both DLLs must be distributed and registered together in the target machine.

There are two functions added to the custom function list: `Encode` and `Encode2`. Function `Encode2` allows you to adjust the target size as well as the line feed separator. Under most cases, `Encode` is sufficient.

```

MoroviaPDF417Encode( BSTR strDateToEncode, LONG trunk_no, BSTR*
strOutput )

```

Note that Crystal Reports do not support formula field with length more than 255 characters. To get around this limit, the encode functions take an additional parameter

called `trunk_no`. If the result barcode string is longer than 200 characters, you need to call this function multiple times. Each trunk holds maximum 200 characters.

A Crystal Reports sample can be found under PDF417Fontware folder. For more information about how to use Morovia PDF417 Fontware in Crystal Reports, visit <http://www.morovia.com/font/support/crystal-reports-ufi-PDF417.asp>.

DLL API Interface

The encoder ActiveX DLL, PDF417FontEncoder.dll, also exposes a DLL interface to facilitate the application integration with programs only support DLL interface. The DLL API contains one function:

```
LONG DllPDF417FontEncode (
    LPCSTR lpszData, // data to be encoded
    LONG PDFMaxCols, // number of columns (1-30; 0
    automatic)
    LONG PDFMaxRows, // number of rows (3-90, 0 automatic)
    LONG PDFSecurityLevel, // security level (0-8, 9
    automatic)
    BOOL PDFTruncatedSymbol, // Truncated version?
    LPCSTR lpszLineFeedString, // The line separator
    LPSTR szOutput, // output string buffer
    LPDWORD dwLen //the length of the output
);
```

Parameters

lpszData

[in] Pointer to a null-terminated string that specify the data to be encoded. If a partial string has the form `\ddd` (a back slash followed by three digits) it converts to the character which corresponds to the three digit value.). For example, `\013` is converted to carriage return (`\r`). To encode the back slash `\`, use `\\`.

PDFMaxCols

[in] The number of columns of the PDF417 symbol generated. Can be any value between 1 to 30. If you choose value 0, the encoder automatically picks the best value of the number of columns(recommended).

PDFMaxRows

[in] The number of rows of the PDF417 symbol generated. Can be any value between 3 to 90. If you choose value 0, the encoder automatically picks the best value of the number of rows (recommended).

PDFSecurityLevel

[in] The security level of the PDF417 symbol generated. Can be any value between 0 to 8. If you choose value 9, the encoder picks up the security level based on data encoded (recommended).

PDFTruncatedSymbol

[in] Determined whether the PDF417 symbol generated is the normal version or the truncated version. 0 – normal version, 1 – truncated. Value 0 is recommended for this property.

lpszLineFeedString

[in] A string specifying the line separators. At default it is “\r\n” (carriage return plus line feed). Alter this parameter only if needed.

szOutput

[out] Pointer to a buffer that receives the barcode string. Must not be NULL.

dwLen

[in, out] Pointer to a variable that specifies the size, in bytes, of the buffer pointed to by the szOutput parameter. When the function returns, this variable contains the size of the data copied to szOutput. Note that the size reported does not include the ending NULL character.

```
LONG DllPDF417FontEncode (
    LPCTSTR lpszData, // data to be encoded
    LONG PDFMaxCols, // number of columns (1-30; 0
automatic)
    LONG PDFMaxRows, // number of rows (3-90, 0 automatic)
    LONG PDFSecurityLevel, // security level (0-8, 9
automatic)
    BOOL PDFTruncatedSymbol, // Truncated version?
    LPCTSTR lpszLineFeedString, // The line separator
    LPTSTR szOutput, // output string buffer
    LPDWORD dwLen //the length of the output
);
```

The following code snippet demonstrates the use of the DLL API:

```
Char data[]="MOROVIA";
Char results[1024];
int dwLen = 1024;
DllPDF417FontEncode(data, 0, 0, 9, FALSE, "\r\n", results,
&dwLen);
```

The earlier version of this document told that you should register this ActiveX DLL in order to use the DLL interface. This was incorrect. If you write your own application

based on the DLL interface you do not need to register the ActiveX DLL in the target machine. Other programs and samples come with the PDF417 fontware package do require this ActiveX dll to be registered with system. To register the ActiveX, type the following command under the command prompt:

```
RegSvr32 PDF417FontEncoder.dll
```

Index

code snippet, 7

Crystal Reports, 2, 5

DllPDF417FontEncode, 5, 7

DllPDF417FontEncode, 7

GUI, 2, 3, 2

Morovia, 7

MoroviaPDF417Encode, 5

PDF417, 0, i, 1, 2, 3, 2, 3, 4, 5

PDF417FontEncoder.dll, 3, 4, 5, 7

Visual Basic, 4