
Harlequin RIP™

Font Encryption with HQcrypt

Technical Note Hqn 052

December 2009



GLOBAL GRAPHICS®

1 Introduction

HQcrypt from Global Graphics® is a utility for encrypting fonts and PostScript® language files to be run on the Harlequin® RIP. As far as we know the encryption scheme used in HQcrypt has not been broken. The RIP can successfully decrypt a font that has been encrypted with either an OEM number, which is the same for all RIPs supplied by an OEM, or a Harlequin RIP serial number, which is different for each RIP.

HQcrypt runs on the following platforms:

- MS-DOS®
- Windows® (from a DOS command prompt).
- Macintosh® OS X.

1.1 Contents

This document contains the following sections:

- “Command line for HQcrypt”—See Section 2 on page 1.
- “Procedure for encrypting fonts”—See Section 3 on page 2.
- “Interaction with basic font format”—See Section 4 on page 3.
- “Strategies for supplying encrypted fonts”—See Section 5 on page 8.

2 Command line for HQcrypt

HQcrypt is executed from a command line, taking in a file to encrypt and a Harlequin RIP serial number or OEM number to encrypt against, and outputting an encrypted file.

On a Windows PC the general command line format is:

```
type file_in.ps | hqcrypt.exe <serial_number|OEM_number> >
file_out.ps
```

For example

```
type OrigFont\Helvetica | hqcrypt.exe 12345 > EncFont\Helvetica
```

On Macintosh OS X use the command line:

```
cat file_in.ps | ./hqcrypt <serial_number|OEM_number> >
file_out.ps
```

For example

```
cat OrigFont/Helvetica | ./hqcrypt 12345 > EncFont/Helvetica
```

2.1 Universal and evaluation RIP serial numbers

As well as OEM and RIP specific serial numbers, HQcrypt also supports two special serial numbers. These are **59** for *any* Harlequin RIP and **11** for Evaluation RIPs. For example

```
type OrigFont\Helvetica | hqcrypt.exe 11 > EncFont\Helvetica
```

would allow the encrypted font to be used on an evaluation copy of the Harlequin RIP.

3 Procedure for encrypting fonts

HQcrypt is generally used to encrypt fonts. Encrypted fonts are placed in the RIP's font folder and are decrypted automatically by the RIP when required, without the need for user intervention. Encrypted fonts remain encrypted in the fonts folder, and are therefore secure at all times.

The following procedure describes how to encrypt fonts with HQcrypt.

1. Decide if you are encrypting fonts for use on all Harlequin RIPs supplied by an OEM, or just for one particular RIP serial number. If you are encrypting for all RIPs supplied by an OEM, you will need to know the OEM number, which can be obtained from Global Graphics on demand, or directly from a dongle itself—it is the first two hexadecimal digits of the fourth part of the dongle part number (which is on the dongle label). So, for example, if your dongle part number is ME1-N-12-0BD-0-ALL, then your OEM customer number is 0B (hexadecimal), which is 11 decimal. Serial numbers are determined from one of several sources:

1. Determine which format the fonts will be in when encrypted. Type 1 fonts should be converted to DLD1 before this stage, see Section 4 on page 3. If necessary determine which files of a Type 4 font require encryption.
2. Move the fonts to be encrypted into a temporary directory or folder, which we will refer to as **OrigFont** on the Macintosh or PC. Create a second directory or folder into which the encrypted fonts are to be placed. We will call this **EncFonts**.

On a PC you should type the following under DOS:

```
type OrigFont\Helvetica | hqcrypt.exe 12345 > EncFont\Helvetica
```

On a Mac OS X you should type the following in a terminal application:

```
cat OrigFont/Helvetica | ./hqcrypt 12345 > EncFont/Helvetica
```

Note that you must leave the name of the file unchanged, as the name of the destination file must be the name of the font, or the mapped name of the font, for it to work correctly.

3. You now have a folder of encrypted fonts that can be copied around as required and shipped to the customer who has the corresponding dongle, or any customers of a particular RIP vendor if you are encrypting using the OEM number.
4. To use these encrypted fonts, copy them into the **sw/fonts** folder of the RIP and install the dongle that they were encrypted for. The RIP will automatically decrypt the fonts as required, ensuring that the fonts are never available in decrypted form. The RIP will only be able to decrypt the fonts if the correct dongle is installed.

4 Interaction with basic font format

The RIP is capable of understanding a proprietary font format known as DLD1 format, to which Type 1 font files may be converted. DLD1 format fonts can consume as little as 2k of RIP VM per font, and offer improved performance over Type 1 and Type 4 fonts. Global Graphics recommends conversion

of all Type 1 fonts to DLD1 on installation in the RIP. Integrated Harlequin RIP products on the Mac and PC do this automatically during installation of Type 1 Latin fonts.

Examination of a Type 1 font is not a very complex process, and a competent PostScript language programmer could write code to process any Type 1 font in virtual memory and dump it to a file in some structured way. This applies whether the Type 1 font is held in an encrypted file in the RIP's file system or not. It is possible to discover everything about a Type 1 font by examining the PostScript language structures that define the font after it has been loaded using `findfont`. The possibility of examining the PostScript language structure of a font using a PostScript language program and of creating a PostScript language file in the RIP's file system which, if interpreted, would reproduce the font's structure and thus reproduce the font itself applies to all Type 1 fonts on all PostScript language compatible interpreters (including those produced by Adobe), regardless of the installation techniques employed. Once the font is in the PostScript language world, a program can examine it and dump it to a file. The DLD1 format prevents this, by not loading the character definitions into the PostScript language world at all, so that such a dumping program would produce a copy of the font with all the important parts missing. Therefore Global Graphics recommends DLD1 conversion as the first stage of the encryption process, because without it, Type 1 fonts are not strongly protected.

If a font is to be encrypted, and it is to be converted to DLD1 format, the conversion to DLD1 format *must* precede the encryption stage.

To convert Type 1 fonts to DLD1 format they may be installed into a RIP using the Install Font menu option on a RIP with a GUI front end, by downloading over AppleTalk using FireWorks, or through the `-download` option on a Unix command-line RIP. If this is not possible (for example, for some leaf fonts of composite fonts) then you may convert the files using the following two steps:

- Copy the Type 1 fonts to the `SW/fonts` directory in the RIP.
- Interpret the file `SW/Usr/Optimize Fonts (SW\USR\OPTIFONT.PS` on an NT RIP).

All these comments about Type 1 fonts apply equally to those Type 1 fonts that are base or "leaf" fonts of a composite (Type 0) font.

Type 3 font programs can also be encrypted, after minor modifications, but once loaded into virtual memory their entire structure can be examined trivially, regardless of any protection method attempted, so they are not strongly protected.

4.1 Options for encrypting Type 4 fonts

The RIP will successfully decrypt a font file only if it is accessed using the **findfont** operator or its equivalent **selectfont** and **findresource** operators. Thus such a protected font file cannot usefully be read even by PostScript language file operations; the decryption will not occur. These considerations apply equally to those Type 1 and Type 4 fonts that are base or leaf fonts of a composite (Type 0) font and those which are Latin, non-composite, fonts.

Type 1 fonts converted to DLD1 consist of a single file, which is executed using **findfont**, and which can therefore be encrypted and made completely secure.

The Type 4 font format is a less efficient way of solving the problem that DLD1 format addresses – reducing the amount of virtual memory that a font consumes. Type 4 fonts typically consist of a PostScript language program file that is executed using **findfont** as usual, plus one or more data files, and sometimes other common runtime PostScript language program files for procedures or data that are shared between many Type 4 fonts. This is particularly common in the case of Type 4 fonts that are leaf fonts of a composite font. The data files typically contain font outline data and offset tables to facilitate access to it. The **BuildChar** procedure of the Type 4 font uses PostScript language file operators to read a chunk of data from the font outline file and then execute it to build the character bitmap using a special operator called **CCRun**. The common runtime files are usually just **run** by the Type 4 font program.

These arrangements pose some problems for secure encryption of Type 4 fonts. The data files are accessed using general PostScript language procedures using file operators. If the data were encrypted, these routines would not decrypt it, and the character outline data would be nonsense. If these routines were to note that the files were encrypted and decrypt the data automatically, or a file mode or filter could be used to cause automatic decryption, then any encrypted font could be copied using a PostScript language pro-

gram. Thus it is not feasible to encrypt the font outline data securely. The run time files are executed using an explicit `run` invocation, which does not decrypt an encrypted file, and which will stop immediately with a PostScript error if presented with an encrypted file to run. Thus these files cannot be securely encrypted.

It is worth mentioning that even if it were possible to encrypt all files associated with a Type 4 font securely, all the font data could be accessed by redefining PostScript language operators to dump the data in clear format to a different file, and by writing code to traverse the font dictionary structure and dump information to a file; the font could be recreated in clear format by a skilled PostScript language programmer.

There are several options available regarding encryption and protection strategies for Type 4 fonts that are to be bundled with a Harlequin RIP product.

- Obtain Type 1 versions, convert to DLD1 and encrypt.

Font vendors usually use some standard font design tool which keeps the font data in its own internal format, and which can generate a choice of font formats on demand. It would not be technically difficult for most font vendors to produce Type 1 versions of fonts that they normally produce in Type 4. This is the only option that provides complete security, and so the font vendor may be happy to assist with such a request. Therefore Global Graphics strongly recommends this option where it is possible.

- Encrypt only the parts of the Type 4 fonts that are run using `findfont`.

This option requires the RIP vendor to inspect the fonts, work out which files are interpreted with `findfont` and encrypt those. This usually means the files in the fonts directory, but in some examples there are files in the fonts directory that are run directly. The files can be sorted into those for encryption and those that are run by judicious use of a string searching tool such as `grep`, and looking for `findfont` to determine files that can be encrypted, and `run` for those that cannot.

Encrypting only parts of the font is a fairly secure subterfuge, but not completely secure because the font structure can always be unpicked using a PostScript language program enabling its re-creation by a skilled programmer.

- Obscure the outline data and encrypt all PostScript language program files.

This option requires the RIP vendor to modify the font programs, in a very straightforward way, to use a special operator instead of `run` to execute the common and runtime files that normally cannot be encrypted. The special operator is in `internaldict`. This step increases the obfuscation. It also protects the procedures within the font, and so the font outline data may also be obscured by some means, being decoded in the `BuildChar` procedure within the now somewhat secure font. This method is still not completely secure because the font structure can always be unpicked using a PostScript language program, enabling its re-creation by a skilled programmer.

In summary, Global Graphics Font Encryption can be used to obscure Type 4 fonts considerably, but not to make their font data completely secure. This is done by selectively encrypting some of the font files, or taking further steps to use the decrypting `run` operator on all PostScript language files in the font explicitly. In any case, some understanding of the working of the font is required of the RIP vendor, assuming it is they who do the encryption. The analysis of the font is quite straightforward.

A Type 4 font will never be completely secure because the secrets that provide its protection are held entirely in the PostScript language world, and can therefore be unpicked by a determined programmer, unless a font vendor works closely with Global Graphics to implement an additional encryption scheme within the RIP.

The only way to achieve complete security is to acquire a Type 1 version of the font, convert it to DLD1 format and encrypt that. In this case the secrets stay inside the RIP software and are inaccessible to the PostScript language world, and so the font is highly secure.

Of course, for a composite font, the Type 0 root font should be encrypted regardless of the types of the leaf fonts. (If it is not encrypted, it is possible to find out which glyphs are in which leaf fonts.)

5 Strategies for supplying encrypted fonts

There are several options available regarding encryption and protection strategies:

- The RIP vendor supplies fonts keyed to individual RIPs.

The RIP vendor must have a clear format version of the font. The RIP vendor converts fonts to DLD1 format as appropriate, then encrypts the resulting collection of files to an individual customer's RIP ID on receipt of order.

The RIP ID is available from the customer's license file or dongle, which the RIP vendor will have kept in a database for maintenance reasons. The RIP vendor needs font encryption tools, available from Global Graphics.

The installation procedure is a binary file copy into the fonts folder in the RIP's file system. Installation tools are trivial.

This is very secure, because the fonts supplied to the customer will only work with their particular RIP, with their particular license or dongle. In addition, the customer never has a clear format copy of the font, even in transit in an install program.

- The font vendor supplies fonts keyed to individual RIPs.

The font vendor has a clear format version of the font, and converts fonts to DLD1 format as appropriate. A copy of the RIP is required for this. The font vendor then encrypts the resulting collection of files to an individual customer's RIP ID on receipt of order. The RIP ID is available from the customer's license file (which the customer can read but not

alter) or from the RIP System Monitor. The font vendor needs the Harlequin RIP and Global Graphics encryption tools, perhaps supplied by the RIP vendor.

The installation procedure is a straight-forward file copy into the fonts folder in the RIP's file system. Installation tools are trivial.

This is very secure, because the fonts supplied to the customer will only work with their particular RIP, with their particular license or dongle. The customer never has a clear format copy of the font, even in transit in an install program and neither does the RIP vendor, so the very protective font vendor will be happy.

From the customer's viewpoint, these methods are equally secure. For this level of security, the RIP ID information must be folded into the font data, before it reaches the customer. Someone must do this, either the RIP vendor or the font vendor.

- The font vendor supplies fonts keyed to a particular vendor's RIP.

The font vendor has a clear format version of the font, and converts fonts to DLD1 format as appropriate. A copy of the RIP is required for this. The font vendor then encrypts the resulting collection of files to the RIP vendor's Global Graphics OEM Number. The encrypted files can be sent to all customers with the Harlequin RIP from this particular RIP vendor. The font vendor needs the Harlequin RIP and Global Graphics encryption tools, perhaps supplied by the RIP vendor.

The installation procedure is a binary file copy into the fonts folder in the RIP's file system. Installation tools are trivial.

This is only fairly secure against unauthorized use of the font, because the supplied font files will work with all Harlequin RIPs supplied by the particular RIP vendor. It is very secure against theft of the font data by other font houses, since the font data are encrypted and only the Harlequin RIP can decrypt it.

- The RIP vendor supplies fonts keyed to their own RIP.

The comments here are much like those for the previous option, except that the RIP vendor must have a clear format copy of the fonts, and the font vendor need not have the RIP or the encryption tool.

These two methods allow a customer with several copies of the RIP from the RIP vendor to buy only one copy of the font and use it several times without paying, simply by copying the files to the other RIPs.

5.1 Other notes

DLD1 conversion need not be performed using exactly the RIP that the RIP vendor sells; a Macintosh or PC Demo version of Global Graphics' integrated RIP product could be used to do the job, and be protected by a dongle. Please contact Global Graphics Support to discuss the matter further if required.

The DLD1 font converter cannot convert encrypted Type 1 fonts to DLD1 format, because it runs the files it is instructed to convert and the PostScript language `run` operator does not understand encrypted files. Running the files loads the fonts into virtual memory just as `findfont` does.

Change history		
Draft	2002.09.17	First draft
v 1.0	2002.10.16	First released version
v 1.1	2009.12.03	Remove Mac Classic info



Issue 101

Copyright © 1992–2009 Global Graphics Software Limited.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Global Graphics Software Limited.

The information in this publication is provided for information only and is subject to change without notice. Global Graphics Software Limited and its affiliates assume no responsibility or liability for any loss or damage that may arise from the use of any information in this publication. The software described in this book is furnished under license and may only be used or copied in accordance with the terms of that license.

ScriptWorks is a registered trademark and Harlequin, the Global Graphics Software logo, Harlequin RIP, ColorPro, EasyTrap, FireWorks, FlatOut, Harlequin Color Management System, HCMS, Harlequin Color Production Solutions, HCPS, Harlequin Color Proofing, HCP, Harlequin Error Diffusion Screening plugins 1 bit and 2 bit, HEDS1 and HEDS2, Harlequin Full Color System, HFCS, Harlequin ICC Profile Processor, HIPP, Harlequin Standard Color System, HSCS, Harlequin Chain Screening, HCS, Harlequin Dispersed Screening, HDS, Harlequin Micro Screening, HMS, Harlequin Precision Screening, HPS, HQcrypt, Harlequin Screening Library, HSL, Harpoon, RipFlow, ScriptWorks MicroRIP, ScriptProof, ProofReady, Scalable Open Architecture RIP, SOAR, SetGold, SetGoldPro, TrapMaster, TrapPro, TrapProLite, TrapWorks, PDF Creator and RIPFlow are all trademarks of Global Graphics Software Limited.

Portions licensed under U.S. Patents: Nos. 4,500,919, 4,941,038 and 5,212,546. EasyTrap is licensed under one or more of the following U.S. Patents: Nos. 5,113,249, 5,323,248, 5,420,702, 5,481,379.

Adobe, Adobe Photoshop, Adobe Type Manager, Acrobat, Display PostScript, Adobe Illustrator, PostScript, Distiller and PostScript 3 are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries which may be registered in certain jurisdictions.

Global Graphics Software Limited is a licensee of Pantone, Inc. PANTONE® Colors generated by ScriptWorks are four-color process simulations and may not match PANTONE-identified solid color standards. Consult current PANTONE Color Publications for accurate color. PANTONE®, Hexachrome®, and PANTONE CALIBRATED™ are trademarks of Pantone, Inc. © Pantone, Inc., 1991.

Other brand or product names are the registered trademarks or trademarks of their respective holders.



US Government Use

The ScriptWorks software is a computer software program developed at private expense and is subject to the following Restricted Rights Legend: "Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in (i) FAR 52.227-14 Alt III or (ii) FAR 52.227-19, as applicable. Use by agencies of the Department of Defense (DOD) is subject to Global Graphics Software's customary commercial license as contained in the accompanying license agreement, in accordance with DFAR 227.7202-1(a). For purposes of the FAR, the Software shall be deemed to be 'unpublished' and licensed with disclosure prohibitions, rights reserved under the copyright laws of the United States. Global Graphics Software Incorporated, 95 Sawyer Road, Waltham, Massachusetts 02453."

