Harlequin® RIP

Encryption Library

Technical Note Hqn 056

February 2004

GLOBAL GRAPHICS®

1 Introduction

HQXLIB provides on-the-fly encryption for software passing information into the Harlequin RIP. The Harlequin® RIP™ has previously used the HQcrypt utility to secure font information passed into the RIP. See Technical Note Hqn 052 for more information. The resulting encryption produced by HQcrypt and HQXLIB is referred to as "HQX-encrypted".

HQcrypt provides one-time output for the RIP to use in the future. For use in real time applications, the static library HQXLIB has been developed to integrate into OEM plugins or applications that may want to communicate with the RIP securely. This currently relates to font information and threshold screens (`/Encrypt` dictionary at `sethalftone`).

HQXLIB is supplied in two forms:

- Win32 single-threaded static library
- Win32 multi-threaded static library

2 Overview

`hqxlib.lib` has two exported functions:

```
int hqx_expected_length(int);

int hqx_encrypt(unsigned char * , int , int );
```

3 Example

In this example, the original data that needs to be encrypted is `x` bytes long.

To establish how much memory is allocated for the HQX-encrypted buffer, call `hqx_expected_length(x)` to get the length of the buffer (`buf`) that needs to be allocated. In this example it is `y` bytes long.

```
int y = hqx_expected_length( x );
```

You may want to check that `y > x`. This will always be the case. Although `(y - x)`, the amount of additional bytes needed, is subject to change.

Put the original data that needs to be HQX-encrypted at the start of the newly created buffer that is `y` bytes long.

Choose whether you are HQX-encrypting the buffer for a specific RIP (RIP ID), for any OEM RIP (the OEM number), or for the generic HQX-encryption (59), and pass this number in as the key. For more information, see section 3.0, step 1 from Technical Note Hqn 052.

In this example;

```
int id = 59;
```

Pass the buffer, the original length of the buffer and the selected key into the `hqx_encrypt` function.

```
int length_out = hqx_encrypt( buf, x, id );
```

If `length_out < 0`, an error has occurred.

Typical errors include the `buf` being `NULL` or `x < 1`.

`length_out <= x` is also problematic. This sort of error should be referred to Global Graphics Harlequin RIP Support with example code.

The buffer (`buf`) is now HQX-encrypted of length, `length_out`.

In almost all applications, `length_out == y`. This may be something to check for within debug environments too.

4 Threshold data protection

There is an optional dictionary for `sethalftone` called `/Encrypt`.

Here is an example:

```
<<
    /HalftoneType         16
    /Width                20
    /Height               40
    /Encrypt    <<
                /EncryptType              1
                /EncryptLength            1719
    >>
    /Thresholds  currentfile    /ASCIIHexDecode  filter
>>  sethalftone

585865654B4B0B0B4E4EF9F983831616EDED090923239595D6D63333B7B7...
```

The HQX-encrypted Halftone at `sethalftone` can be used with Type 6, Type 10, and Type 16 Halftones.

The `/Encrypt` dictionary currently contains two entries;

`/EncryptType`
: has to be set to `1` currently (`1` = HQX-encrypted*).

`/EncryptLength`
: the length in bytes of the HQX-encrypted data. The HQX-encrypted threshold is always longer than the natural threshold.

* HQX-encrypted refers to encrypted output from either HQcrypt or HQXLIB.

To obtain the encrypted halftone threshold, take the natural threshold (that is, pre-ASCIIHex and so on), and run HQcrypt on it with the desired key (number) as described in Technote 052, Section 3.0. Alternatively, HQXLIB can be used to obtain the HQX-encrypted threshold.

Note: You will find `hqcrypt.exe` in the `\RIPs\PC\<version>\<revision>` folder. For example, under Windows NT you will find it in the, `\RIPs\PC\win_32-pentium\all` folder for the version of the GUI RIP.

Change history		
v 1.0	2004.02.05	New Document

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

US Government Use

<ProductName> 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, 5875 Trinity Parkway, Suite 110, Centreville, VA 20120.