

---

---

# Integrating and configuring screening modules for the GUI RIP

Technical Note Hqn076

January 2008



# 1 Introduction

This document describes how to integrate and configure screening modules for the GUI RIP. A number of screens are supplied with the Harlequin RIP. It is possible however to add custom screening modules. These must be integrated using the following procedures.

## 1.1 Example modules

In order to illustrate the installation and configuration of a screening two examples are provided in the Plugin kit. You will find them here:

`\HarlequinRIP_X_X_Generic\PluginKit\HTMScreening\examples\src`

<b>htm4x4</b>	This file provides an example of a bi-level halftone. This could not be produced using PostScript language thresholds because, when moving from one shade to a darker shade, pixels can be extinguished as well as lit.
<b>htm8ml</b>	This file provides an example of threshold tables. Threshold tables can be thought of as very small, square, HDS cells (Harlequin Dispersed Screening). However, they are not used as a simple threshold screen, but as a distribution table. A contone level above or below the required level is chosen according to the value in the table aligned on the raster grid.

These files are provided to illustrate the various possibilities available with screening modules. Better, more efficient methods can be achieved.

**Note:** Where X\_X is the version number of the release.

## 2 Installing the screen

There are several stages to preparing a screening module:

- [“Adding the screen to the core RIP” on page 3.](#)
- [“Configure the screen for use with the GUI RIP” on page 4.](#)
- [“Configure the screen for use with specific output plugins” on page 5.](#)

## 2.1 Adding the screen to the core RIP

This process make the screen available to the core RIP but it cannot be used by the GUI RIP until it is given a **Dot shape** name, see below...

The new screen must be added to the RIP screens list, which specifies a particular screen configuration. Adding the screen to the RIP is done using the `InstallHalftoneDef` function.

The simplest invocation for adding a monochrome screen is:

```
/EgHalftone <<
  /HalftoneType    100
  /HalftoneName    /EgHalftone
  /HalftoneModule  /htm8ml
  /Screen 0
  /Levels 2
>>
/HqnHtm /ProcSet findresource /InstallHalftoneDef get exec
```

**Note:** The **HqnHtm** plugin procset is for screening only.

For the provided examples (**htm8ml** and **htm4x4**) the parameters `/Screen` and `/Levels` are required, but other screen modules might require parameters with different names or no parameters at all. The function `m18HalftoneSelect` (in **htm8ml.c**) contains an array that is parsed for option names. `SW_DATUM_BIT_NOthing` within the array is a flag which defines the option `/Inverse` as being optional. Its absence indicates that the parameter is required.

A more useful invocation, which allows the individual tailoring of the screens on a per colorant basis (required for multi-colorant use of **htm8ml** and **htm4x4**) is:

```
/EgHalftone <<
  /HalftoneType    5
  /HalftoneName    /EgHalftone
  /Cyan    << /HalftoneType 100 /HalftoneModule /htm8ml /Screen 0
              /Levels 2 >>
  /Magenta << /HalftoneType 100 /HalftoneModule /htm8ml /Screen 1
              /Levels 2 >>
  /Yellow  << /HalftoneType 100 /HalftoneModule /htm8ml /Screen 2
              /Levels 2 >>
>>
/HqnHtm /ProcSet findresource /InstallHalftoneDef get exec
```

The `InstallHalftoneDef` function links a name that can be used with `setscreen` to the name of a screening module.

**Note:** `HalftoneType 5` defines a composite screen containing an arbitrary number of halftone screens, one for each color component. The keys in this dictionary are the names of color components (CMYK or spot) The values are halftone dictionaries of other types, each defining the halftone screen for a single color component.

The PostScript language code must be run at startup. For more information see [“Running the PostScript language start up code” on page 9](#).

## 2.2 Configure the screen for use with the GUI RIP

Once the screen is available to the core RIP it must now be configured for the GUI RIP by creating a specific screen configuration under a unique name, called the **Dot shape**. To do this an entry needs to be added to the GUI RIP **Screen Names** file.

**Note:** The **Screen Names** file only appears after the RIP has been started-up at least once.

1. Open the file:  
`/SW/Config/Screen Names`
2. Append your Screen Names definition after the last entry in the file. This is an example Screen Names definition:

```
<<
/InternalName (EgHalftone)
/ExternalName (Example Halftone)
/Enabled true
/DefaultValues []
>>
```

Change `InternalName` to your core screening module name within the `sw_htm_api` structure. This name must be unique within each RIP instance.

The `ExternalName` string is displayed in the **Dot shape** field of the “Edit Style for Halftone separations” dialog in the RIP GUI. The `ExternalName` does not have to be unique within the RIP instance, but it could cause confusion if it is not. If several arrays are added, the order they appear in the file determines the order in which they appear in the **Dot shape** field in the GUI.

## 2.3 Configure the screen for use with specific output plugins

If you are implementing a screen for an output device that includes its own Screen Names file, the new screen must also be added to that. Therefore, the new Screen Names file must be included in that device. These files are located in the actual output plugin. For example:

**\HarlequinRIP\_X\_X\_Generic\SW\Devices\epson4800\Screen Names**

There may be a number of files within the device Screen Names directory, all of which may be updated with the new entry. These files may exist because there might be limitations to the screening the device can support. For example, VSD printers use 2-bits per pixel screening and most standard screens cannot cope with that. Therefore, a HEDs file must be provided that copes with this and omits the other standard screening set (round, euclidian and so on).

If there is a **Screen Names** file entitled with the name of the device it is used. For example:

**\HarlequinRIP\_X\_X\_Generic\SW\Devices\epson4800\Screen Names\SP4800 VSD Sheet**

If not, and a **Screen Names** file exists within the **\Screen Names** directory then that file is used. For example:

**\HarlequinRIP\_X\_X\_Generic\SW\Devices\epson4800\Screen Names\Screen Names**

If neither of these files exist the RIP **Screen Names** file (in **SW\Config**) is used.

In v8.0r0 (and later) RIPs you can add the entry to the **Screen Names** file (in **SW\Config**) by using the `HqnPluginConf*` procset function `AddScreenNames`.

---

\*. `HqnPluginConf` is a general plugin configuration procset though currently it is only used for screening.

The following PostScript language code, which should go into the **ExtraStart** file, shows an example of adding the entry (shown above) using `HqnPluginConf`:

```
[
  <<
    /InternalName (<internal name>)
    /ExternalName (<dot shape name>)
    /Type (Threshold)
    /Enabled true
    /DefaultValues []
  >>
]
/HqnPluginConf /ProcSet findresource /AddScreenNames get exec
```

**Note:** This template has no error checking. See below for a more complete version.

Where `<internal name>` is the key name identifying the screen the core RIP `scrnames` array. `<dot shape name>` is the name used in the Edit Screen styles dialog.

Type can be one of `Threshold`, `SpotFunction` or `HDS`. For core module screens use `/Type (Threshold)`.

For more information regarding the `/DefaultValues` array see “Adding Extra dot shapes” in the document entitled: “Extensions Manual—For the Harlequin PLUS Server RIP”.

To add the `htm8ml` module, found in the Plugin kit, this PostScript language code, which include the error checking omitted earlier, should be placed within

**\HarlequinRIP\_X\_X\_Generic\SW\Extensions\CoreModule\htm8ml\ExtraStart.**

```

[
  <<
    /InternalName (EgHalftone)
    /ExternalName (Example Halftone)
    /Type (Threshold)
    /Enabled true
    /DefaultValues []
  >>
]
/HqnPluginConf /ProcSet resourcestatus
{
  pop pop
  % pass an array of Screen Names to be merged with the default one
  /HqnPluginConf /ProcSet findresource /AddScreenNames get exec
}
{
  pop
  (%stderr%)(w) file (%%[HqnPluginConf is not available, so Core   htmeg
has not been added to standard plugins. %%]\n)   writestring
} ifelse

```

If required, the array passed to `AddScreenNames` can contain several entries which will be added, in the order presented, to the end of the existing array. The PostScript language code to do this can be added to the same **ExtraStart** file that includes `InstallHalftoneDef`.

**Note:** For a screen names entry where the `type` is set to `early screen` (that is, Javelin) the RIP will not enable the screen name (and hence the module) even if the file is set to `true`. This prevents the GUI from presenting options that are not actually available. The module can be enabled via a PostScript language file (but in the case of Javelin it is still not there so will not work).

Once installed, the plugin will report its presence in the RIP monitor when the RIP is booted. For example:

```
"Core Screening Example Plugin 1"
```

```
Version 1.0r1 - Copyright (c) 1998-2007 Global Graphics Software Ltd. All
Rights Reserved."
```

The examples provided do not require a password, but if you change an example to indicate that a password is required, you must use the **Harlequin RIP > Configure RIP** menu item, followed by the **Extras...** button. In the list displayed there will be an entry in the

form: <Example Name>, <Example Security Name>. Click on this and enter the required password. Ensure that the password successfully enables the plugin by checking that the enabled field displays *yes*.

To use the new screening module a halftone page setup must be created. Open the **Dot shape** field of the “Edit Style for Halftone separations” dialog in the RIP GUI. This is where you would usually see items such as *Round* and *Euclidean*. If you followed the example above and named the screen using the *ExternalName* in the Screen Names entry, there should be an additional entry *Example Halftone*. When completed you should RIP a job an examine the results.

## 2.4 Important names

This section provides details of the names required for integrating and configuring your screening modules:

- The name of the plugin executable and the directory it is stored in. It is recommended that these are the same, but they need not be. This is required for installation of the module.
- The two names returned from *D\_GEN\_SECURITY\_NAME* which together identify the plugin to the security system. (Not applicable in the example above.) These names are required for creating the correct keys to enable the plugin.
- The name of the module (referenced in *sw\_htm\_api* as */HalftoneModule*) that the plugin provides (for example, the plugin **htmeg** provides the module **htm4x4**). This name is used together with */InternalName* and */ExternalName* to create a screening configuration that can be used in the job.
- The */InternalName* (*/HalftoneName*) of the screen configuration that you define. This name is used together with the */HalftoneModule* and */ExternalName* to create a screening configuration that can be used in the job.
- The Screen Names. That is, the */ExternalName* of the screen configuration presented in the **Dot shape** field of the “Edit Style for Halftone separations” dialog in the RIP GUI. In this example it is (**Example Halftone**). This name is used together with the */InternalName* and */HalftoneModule* to create a screening configuration that can be used in the job.



## 2.5 Running the PostScript language start up code

Whichever style of definition is used, a PostScript language file containing the definition should be added to the **ExtraStart** directory within the screening plugin's folder. For the core module screening module htm8ml you might, for example, have the file:

**\HarlequinRIP\_X\_X\_\SW\Extensions\CoreModule\htm8ml\ExtraStart\htm8mlStartUp**

**Note:** This file needs to be run in a context of `true setglobal` which can be done by putting `currentglobal true setglobal` around the file.

## 3 Document history

Change history		
v1.0	10.01.2008	New Document



## Copyright and Trademarks

Integrating and configuring screening modules for the GUI RIP

Version 1.0: January 2008

Part number: Hqn076

Document issue: 104

Copyright © 2007 Global Graphics Software Ltd. All rights reserved.

Certificate of Computer Registration of Computer Software. Registration No. 2006SR05517

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 Ltd.

The information in this publication is provided for information only and is subject to change without notice. Global Graphics Software Ltd. 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.

Harlequin is a registered trademark of Global Graphics Software Ltd.

The Global Graphics Software logo, the Harlequin at Heart Logo, Cortex, Harlequin RIP, Harlequin ColorPro, EasyTrap, FireWorks, FlatOut, Harlequin Color Management System (HCMS), Harlequin Color Production Solutions (HCPS), Harlequin Color Proofing (HCP), Harlequin Error Diffusion Screening Plugin 1-bit (HEDS1), Harlequin Error Diffusion Screening Plugin 2-bit (HEDS2), Harlequin Full Color System (HFCS), Harlequin ICC Profile Processor (HIP), Harlequin Standard Color System (HSCS), Harlequin Chain Screening (HCS), Harlequin Display List Technology (HDLT), Harlequin Dispersed Screening (HDS), Harlequin Micro Screening (HMS), Harlequin Precision Screening (HPS), HQcrypt, Harlequin Screening Library (HSL), ProofReady, Scalable Open Architecture (SOAR), SetGold, SetGoldPro, TrapMaster, TrapWorks, TrapPro, TrapProLite, Harlequin RIP Eclipse Release, Harlequin RIP Genesis Release, Harlequin MultiRIP, Harlequin Parallel Pages and Harlequin VariData are all trademarks of Global Graphics Software Ltd.

Protected by U.S. Patents 5,579,457; 5,808,622; 5,784,049; 5,862,253; 6,343,145; 6,330,072; 6,483,524; 6,380,951; 6,755,498; 6,624,908; 6,809,839.

Other U.S. Patents Pending

Protected by European Patents 0 803 160; 0 772 934; 0 896 771; 672 29 760.8-08.

Portions licensed under U.S. Patent No. 5,212,546; 4,941,038.

TrueType is a registered trademark of Apple Computer, Inc.

The ECI and FOGRA ICC color profiles supplied with this Harlequin RIP are distributed with the kind permission of the ECI (European Color Initiative) and FOGRA respectively, and of Heidelberger Druckmaschinen AG (HEIDELBERG).

The IFRA ICC profiles supplied with this Global Graphics Software are distributed with the kind permission of IFRA and of GretagMacbeth.

International Cooperation for Integration of Processes in Prepress, Press and Postpress, CIP4, Job Definition Format, JDF and the CIP4 logo are trademarks of CIP4.

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 Ltd. 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

Harlequin RIP 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, Somerset Court, Suite 320, 281 Winter Street, Waltham, MA 02451.