Harlequin Dylan

Remote and Just-in-Time Debugging

Version 1.1 Beta

harlequin™

Copyright and Trademarks

Harlequin Dylan: Remote and Just-in-Time Debugging

Version 1.1 Beta

September 1998

The information in this publication is provided for information only and is subject to change without notice. Harlequin Group plc 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.

US Government Use

The Harlequin Dylan 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 Harlequin’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 For purposes of the FAR, the Software shall be deemed to be `unpublished’ and licensed with disclosure prohibitions, rights reserved under the copyright laws of the United States. Harlequin Incorporated, One Cambridge Center, Cambridge, Massachusetts 02142.”

Europe:

Harlequin Limited
Barrington Hall
Barrington
Cambridge CB2 5RG
U.K.

telephone +44 1223 873 800
fax +44 1223 873 873

North America:

Harlequin Incorporated
One Cambridge Center
Cambridge, MA 02142
U.S.A.

telephone +1 617 374 2400
fax +1 617 252 6505

Electronic Access:

http://www.harlequin.co.uk/
http://www.harlequin.com/

Contents

Remote and Just-in-Time Debugging

This document applies only to the Harlequin Dylan Enterprise edition.

1 Overview of features

The Harlequin Dylan Enterprise edition includes the ability to run, debug, and interact with Dylan applications, DLLs, and OLE controls on remote machines. It also offers a just-in-time debugging facility that allows you to attach the Harlequin Dylan debugger to a Dylan application at the moment of an unhandled error. This document describes how to use these Enterprise-only features.

2 Running a Dylan application on a remote machine

The Harlequin Dylan Enterprise edition offers a facility for running, debugging, and interacting with a Dylan application, DLL, or OLE control ("program" hereafter) running on a remote machine — that is, a networked machine other than the one running the Harlequin Dylan IDE.

The ability to do these things on the remote machine is a simple extension of the standard features described in *Getting Started with Harlequin Dylan*, particularly chapter 6 of that manual which covers debugging and interaction techniques. The techniques for running, debugging, and interacting with a program are exactly the same as for the local machine, but there are a few initial configuration issues to cover.

2.1 Installing the program on the remote machine

To do anything with a Dylan program on a remote machine, the program has to be installed there. It may be there already — if, for instance, you are working with another Dylan developer — but if it is not, you must install the program on that machine.

The simplest way to do this is to open the program's project and use the **Project > Make Release** command, which creates a folder containing the program, any DLLs it uses, and the Harlequin Dylan run-time system DLLs. The folder is called `release` and is created under the project's folder.

1. Choose **Project > Make Release**.

Then you need to take a copy of the debugging information for the run-time system DLLs.

2. In Windows Explorer, go to the `Redistributable` subfolder of the top-level Harlequin Dylan installation folder.
3. Copy the subfolder `Symbols`, and paste it into the project's `release` folder.

Now you are ready to install the program on the remote machine.

4. Copy the `release` folder over to the remote machine.

2.2 Starting the debugging server

Whenever you try to invoke the program on the remote machine, Harlequin Dylan expects to be able to talk to a special debugging server application on the remote machine. If this server is not running on the remote machine, it is not possible to run the Dylan program there either. The debugging server must be started manually on the remote machine itself, so it is worth installing it there.

The debugging server is invoked with **`start-dbg-server.exe`** and is available in the **`Bin`** subfolder of the Harlequin Dylan installation folder. There is also a shortcut to it under **Start > Programs > Harlequin Dylan**. This server calls two other applications, **`nubserv.exe`** and **`rnub.exe`**, so you need to install those too.

1. In Windows Explorer, go to the **`Bin`** subfolder of the top-level Harlequin Dylan installation folder.
2. Copy **`start-dbg-server.exe`** and paste it into a folder on the remote machine.
3. Copy **`nubserv.exe`** and **`rnub.exe`** and paste them into the same folder as **`start-dbg-server.exe`**.
4. Start **`start-dbg-server.exe`**.

If the debugging server starts up correctly, a dialog box appears to confirm that it is running. An icon for it also appears in the Windows task bar. At this point, you can return to the machine running the Harlequin Dylan IDE, where you will be ready to run Dylan programs remotely.

2.3 Starting an application remotely

Running and debugging an application on a remote machine is identical in almost every respect to using your local machine. You can use the three menu items **Application > Start**, **Application > Debug** and **Application > Interact** to launch the application in the normal way. The only difference is that for remote startup you must specify a remote machine in the project's debug settings.

1. Select **Project > Settings...** and select the Debug page.

In the "Remote machine" section, an option box displays the name of the machine on which the project's application is to be run. By default, the selection will be the local machine. Before you can select a remote machine, you must first establish a debugging connection.

2. Click the **Open New Connection...** button.

3. In the dialog that appears, enter the network address of the machine to which you wish to connect, and click **OK**.

   The address is whatever the Windows operating system needs to identify the machine on the network. A computer hostname is likely to be sufficient.

If the connection is successful, the newly connected machine is added to those listed in the “Remote machine” option box. You can now select this machine.

If the connection does not succeed, you should ensure that you have successfully started the debug server program on the remote machine. (See “Starting the debugging server” above.)

4. Select the remote machine in the option box.

Having selected a remote machine, you must also ensure that Harlequin Dylan can locate the program (EXE or DLL) on the filesystem of the remote machine.

5. Specify the path to the program in the “Command line” section of the Debug page.

   The path should be fully qualified, including the name of the remote machine. For example:

   ```
   \\spiral\c_drive\apps\reversi\release\reversi.exe
   ```

   where `spiral` is a machine name, and `c_drive` is the share name of the drive containing the program `release` folder.

You are now ready to run and debug the application remotely. **Application > Start** starts your application running on the remote machine. All the usual debugging and interaction facilities will now be available.

Details about the connection to a remote machine are stored and saved with the project. Harlequin Dylan tries to re-establish the remote connection automatically when you next open and try to run the project.

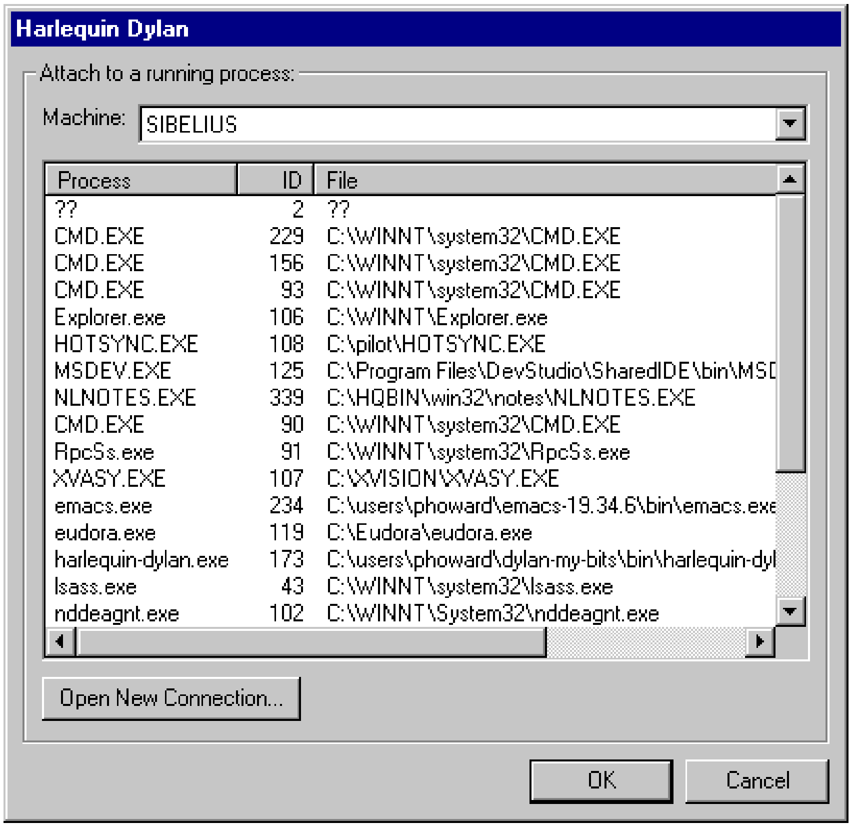
3 Attaching to running processes

The three commands **Application > Start**, **Application > Debug** and **Application > Interact** are all used to launch a program from within Harlequin Dylan. But if the program is running already, perhaps even before you started up the Harlequin Dylan environment, the environment does not know about the process and therefore does not enable the menu and toolbar items for debugging, pausing, stopping, and interacting.

The Harlequin Dylan Enterprise edition allows you to “attach” to such a running process, thereby bringing it under Harlequin Dylan’s control just as if it had been started by the environment. It is very simple to do this:

1. Open the project whose application, DLL, or control is running.
2. From the project window, choose **Application > Attach...**.

   A dialog listing all available running processes appears.



Machine: SIBELIUS

Process	ID	File
??	2	??
CMD.EXE	229	C:\WINNT\system32\CMD.EXE
CMD.EXE	156	C:\WINNT\system32\CMD.EXE
CMD.EXE	93	C:\WINNT\system32\CMD.EXE
Explorer.exe	106	C:\WINNT\Explorer.exe
HOTSYNC.EXE	108	C:\pilot\HOTSYNC.EXE
MSDEV.EXE	125	C:\Program Files\DevStudio\SharedIDE\bin\MSD
NLNOTES.EXE	339	C:\HQBIN\win32\notes\NLNOTES.EXE
CMD.EXE	90	C:\WINNT\system32\CMD.EXE
RpcSs.exe	91	C:\WINNT\system32\RpcSs.exe
XVASY.EXE	107	C:\XVISION\XVASY.EXE
emacs.exe	234	C:\users\phoward\emacs-19.34.6\bin\emacs.exe
eudora.exe	119	C:\Eudora\eudora.exe
harlequin-dylan.exe	173	C:\users\phoward\dylan-my-bits\bin\harlequin-dyl
lsass.exe	43	C:\WINNT\system32\lsass.exe
nddeagnt.exe	102	C:\WINNT\System32\nddeagnt.exe

Figure 1 Selecting a process to attach to.

3. Select the process to which you want to attach, and click **OK**.

After a few moments, the debugger attaches to the running process, and all the normal debugging and interaction facilities become available, just as with **Application > Start**.

Note: Harlequin Dylan does not currently offer any facility for detaching from a process. Once it has been attached to the Harlequin Dylan debugger, and all of your debugging work is finished, you will need to close the program down using the **Application > Stop** command.

The process to which you attach need not be running on the local machine. You can also attach to a process that is running on a remote machine provided that the debugging server application (see “Starting the debugging server” on page 2) is running on that machine. The process list dialog has an option box that allows you to select the machine whose process list you want to view. There is also an **Open New Connection...** button for creating new connections to remote machines, which works in the same way as described in “Starting an application remotely” on page 3.

4 Just-in-time debugging

The Windows operating system has the built-in capability to perform “just-in-time” debugging. This is where a program crashes while not running under a debugger, and the operating system arranges to start up an available debugger and attach it to the crashed process in order to obtain a backtrace. The system registry contains an entry for the debugger that should be invoked when this happens. The Harlequin Dylan Enterprise edition is capable of acting as a just-in-time debugger; during the installation process you have the opportunity to install Harlequin Dylan as your machine’s default debugger.

When the Windows operating system launches Harlequin Dylan as a just-in-time debugger, Harlequin Dylan automatically attaches to the process that has crashed. Once this succeeds, a dialog appears asking whether you would like to open a project. If the crashed process is a Dylan program for which you have a project, you should take this opportunity to open the related project before you start your debugging.

It is not strictly necessary to open a project, since the debugger will still be able to use whatever debugging information is available in the executable file itself in order to provide a backtrace. However, if you do not open a project, you will not be able to use the interactor to evaluate Dylan expressions, nor will you be able to browse the source code or the definitions in the project.

(Of course, if the crashed program is not a Dylan application at all, and also does not contain any embedded Dylan components, then you will not be able to open a project.)

Once you have made your selection and have clicked **OK**, the Harlequin Dylan debugger appears. The debugger provides a full description of the state of the program at the point of the crash. You can proceed with debugging as described in *Getting Started with Harlequin Dylan*.