
Solving Socket connection issues

Technical Note Hqn 066

September 2005



1 Introduction

This technical note describes the working of a socket connection, how the backlog slots affect further incoming connections, and the importance of properly closing connections.

Note: Although this document refers mainly to the Microsoft socket-protocol-stack, the same fundamental description applies to most socket protocol implementations and particularly those based upon the Berkeley sockets API as used in Mac OS X and most Unix and Linux operating systems.

2 Multiple connections

When a program is configured to “listen” to a port using sockets, part of the implicit contract with the socket-protocol-stack is the “backlog” available for other incoming connection requests. Many listening programs can only accept one connection at a time, and until such time as the listener is prepared to accept another connection, the socket-protocol-stack permits incoming connections, but assigns them to one of the “backlog” slots. The next time the listening program is ready to accept another connection, the socket-protocol-stack then gives it the first connection from the backlog.

In the meantime, while the remote program’s connection is in the backlog queue, it will appear to that program as though the connection is complete, and the remote program can send data. In practice, that data just gets queued.

In the normal course of events, the backlogged connection is soon connected, and the queued data gets passed on.

3 Lost data

If a remote program is sending a relatively small amount of data, and impatiently closes the connection very soon after sending it, without waiting for any confirmation of its receipt, the data may be lost before the connection has come out of the backlog.

This type of backlog issue is often the cause of “lost” jobs and “hanging” connections.

The situation can be exacerbated by programs performing untidy socket closures, especially on bi-directional connections, because the untidily broken connection can effectively occupy a backlog slot until such time as the socket-protocol-stack finishes tidying it up.

Once the backlog is filled with queued connections, the stack will reject subsequent connections with a “Connection Refused” message, (code 10061 on Windows, code 61 on Mac OS X, and code 111 on Linux). Of course, “Connection refused” in this case is indistinguishable from “Nobody is listening” (haven’t started RIP or input channels) and “You have a firewall installed”.

The default value of the backlog is implementation dependent. Therefore, it is difficult to be certain what value the protocol-stack will use if the programmer is not specific.

For Windows, Unix/Linux and Mac OS X, Global Graphics use the value 5.

If the first connection is not properly ended, a situation can occur where further connections are accepted but not properly closed and then after that further connections are refused. In this case, the socket-protocol-stack accepts, but sits on, the further connections, until the backlog is filled with pending connections, after which further connections are refused.

By using the common model of “connect-send-disconnect” without a “shut-down” sequence, you would more than likely not see if RIP is hanging on to the connection.

4 Recommended method

A recommended way to properly close-down a connection using the socket APIs is:

1. Indicate to the remote end that you are not planning to send any more data by using the `shutdown` socket-API (preferably non-blocking). For example:

```
shutdown( g_fdSocket, SD_SEND)
```
2. Continue to check for data being sent by the host, until a zero-length read is received. That is, a `select` indicates data is available, but then a `recv` returns `-zero`. This indicates that the far end has finished sending the data, and the shutdown can proceed.

3. Close the socket.

If a program just does the close (step 3) without the proceeding steps, the behavior is socket-protocol-stack dependent as well as being influenced by the `SO_LINGER` socket option. Windows defaults to performing steps (1) and (2) in the background, but does not guarantee to do so without timing-out step (2).

Note: There is no API to confirm that the remote end has received a unit of data which has been sent, without super-imposing an extra protocol known to both parties.

5 Known issues

We have encountered an issue where under certain rare circumstances (which depend upon the state of the PostScript VM allocation mode at the time the interrupt occurs), the RIP may fail to read-off the data from an asynchronous action channel, resulting in the underlying socket connection never being closed by the socket plugin. This in turn will cause subsequent connections to the asynchronous channel to become backlogged, and eventually to be refused.

6 Glossary of terms

Term	Description
API	Application programming interface. In this context we mean the software conventions used to access a particular service or protocol.
non-blocking	A software procedure call which will return to its caller within a very short period of time even if it cannot complete the requested operation immediately. In such cases, the requested operation may either be rejected, or will continue asynchronously, and the return value of the procedure call usually indicates which.
protocol-stack	<p>To allow multiple protocols and services to share access to the network interface hardware, the software components are layered. (This layering usually conforms to the "ISO/OSI 7-layer model").</p> <p>The components in these layers which combine to provide an API to a particular network protocol are commonly referred to as a "protocol-stack". Multiple protocol-stacks can co-exist and exploit the network hardware co-operatively.</p> <p>The protocol-stacks are often provided as part of the operating system, but there are some third-party software protocol-stacks for some operating systems.</p>
socket-protocol-stack	The protocol-stack which provides an API for access to the socket protocol.

7 More information

For more information, see the following:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/closesocket_2.asp

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/graceful_shutdown_linger_options_and_socket_closure_2.asp

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/shutdown_2.asp

Change history		
v1.0	Sept 27, 2005	New Document



Copyright © 1992–2005 Global Graphics Software Ltd.

Document Issue: 101

Copyright © 1992–2005 Global Graphics Software Ltd.

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 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 (HIPP), 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 and Harlequin RIP Genesis Release 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.

Other U.S. Patents Pending

Protected by European Patents 0 803 160; 0 772 934; 0 896 771.

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

LZW licensed under U.S. Patent No. 4,558,302 and foreign counterparts.

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

This 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, 5875 Trinity Parkway, Suite 110, Centreville, VA 20120.