

Graphica Software

160 Queen St

5th Floor

Melbourne, 3000

Victoria, Australia.

e-mail: [sales@melb.graphica.com.au](mailto:sales@melb.graphica.com.au)

© Graphica Software Pty. Ltd. 1997

Developer Note - Using stdin/stdout with Windows

Document: stdiowin.doc

Issue No: Draft

Printed: 05/10/97 16:07

**Table of Contents**

<b>1. Preface</b>	<b>3</b>
<b>2. Key Words</b>	<b>3</b>
<b>3. Related Documents</b>	<b>3</b>
<b>4. Changes History</b>	<b>3</b>
<b>5. Introduction</b>	<b>3</b>
<b>6. Reestablishing stdin, stdout and stderr</b>	<b>3</b>
<b>7. Initialization Counter Ensures stdio Setup</b>	<b>5</b>

**List of Figures****Error! No table of contents entries found.****List of Tables****Error! No table of contents entries found.**

## 1. Preface

The following note provides information on using stdin/stdout/stderr and thus cin/cout/cerr from C++ in Windows 32 bit applications.

## 2. Key Words

WIN32 STDIN STDOUT CIN COUT

## 3. Related Documents

[1] Microsoft Visual C++ Run-Time Library Reference Vol 3 - Active Template Library.

## 4. Changes History

Version 1.0      4 October, 1997              Graphica Software Pty. Ltd.

## 5. Introduction

A standard Windows application does not have the C runtime I/O handles for stdin, stdout and stderr attached.

For code code that was written for a Unix environment that relies on using stdout, stderr and cout and cerr to print out diagnostic and debug information, the lack of support for the standard process runtime file facilities is a nuisance, when porting code from Unix to Windows.

To save having to write special debug code or facilities the simplest solution to this problem is to setup the appropriate file handle as part of the Windows runtime environment.

## 6. Reestablishing stdin, stdout and stderr

The following example shows how to reestablish stdin, stdout and stderr for a Windows Win32 application.

```
/*
 * File: winstdio.h
 *
 * Contents: This file contains a set of 'C' functions that do Win32
 *           specific file handle munging to ensure that stdin, stdout
 *           and stderr are defined to some valid handle for Windows
 *           based applications.
 *
 * Programmer: John Hartley
 *
 * Version: 1.0
 *
 * Date: 25/4/97
 *
 * Version: 1.0
 *
 * License:
 * Copyright (c) 1997 Graphica Software Pty. Ltd.
 * This software is part of the "ObjectBlocks" (TM) reusable components library.
 * Its use is allowed only if this notice is included intact and
 * on the basis of permission being granted by the copyright holder.
 * Graphica Software Pty. Ltd.
 * 851 Brunswick St. North
 * North Fitzroy 3068
 * Victoria, Australia.
 * Ph: + 61 3 9481 3747
 * Fax: + 61 3 9481 3747
 * Email: info@melb.graphica.com.au
 * http://www.graphica.com.au
 */

#ifndef WINSTDIO_H_
#define WINSTDIO_H_

#ifdef __cplusplus
extern "C" {
#endif
```

```
#include "sysioflg.h"

void StdIoConsole(int bOpen);

#ifdef __cplusplus
}
#endif
#endif

/*
 * File: winstdio.c
 *
 * Contents: Create stdio handles by opening a console.
 *
 * Programmer: John Hartley
 *
 * Version: 1.0
 *
 * Date: 24/4/97
 *
 * License:
 * Copyright (c) 1997 Graphica Software Pty. Ltd.
 * This software is part of the "ObjectBlocks" (TM) reusable components library.
 * Its use is allowed only if this notice is included intact and
 * on the basis of permission being granted by the copyright holder.
 * Graphica Software Pty. Ltd.
 * 851 Brunswick St. North
 * North Fitzroy 3068
 * Victoria, Australia.
 * Ph: + 61 3 9481 3747
 * Fax: + 61 3 9481 3747
 * Email: info@melb.graphica.com.au
 * http://www.graphica.com.au
 */

#include <stdio.h>
#include <io.h>
#include <fcntl.h>
#include <windows.h>

#include "winstdio.h"

static int hStdIn = 0,
          hStdOut = 0,
          hStdErr = 0;

void StdIoConsole(int bOpen)
{
    switch (bOpen) {
        case 1: AllocConsole();
                hStdOut = _open_osfhandle((long)GetStdHandle(STD_OUTPUT_HANDLE),
                                         _O_TEXT);
                hStdErr = _open_osfhandle((long)GetStdHandle(STD_ERROR_HANDLE),
                                         _O_TEXT);
                hStdIn = _open_osfhandle((long)GetStdHandle(STD_INPUT_HANDLE),
                                         _O_TEXT);
                *stdout = *fdopen(hStdOut, "w");
                *stderr = *fdopen(hStdErr, "w");
                *stdin = *fdopen(hStdIn, "r");
                setvbuf(stdout, NULL, _IONBF, 0);
                setvbuf(stderr, NULL, _IONBF, 0);
                setvbuf(stdin, NULL, _IONBF, 0);
                break;
        default: FreeConsole();
                break;
    }
}
```

## 7. Initialization Counter Ensures stdio Setup

For C++ it is possible to ensure that the io handles are automatically initialized by using the following initialization counter based static object establishment routines.

```
//
// File: wnioinit.h
//
// Contents: Automatically create a console for
//           stdin, stdout and stderr.
//           Based on the value of the following
//           compiler flags:
//           _STDIO_LOG_ - do init, send to log
//           _STDIO_CONSOLE_ - do init, send to console
//
// Programmer: John Hartley
//
// Version: 1.0
//
// Date: 24/4/97
//
// License:
// Copyright (c) 1997 Graphica Software Pty. Ltd.
// This software is part of the "ObjectBlocks" (TM) reusable components library.
// Its use is allowed only if this notice is included intact and
// on the basis of permission being granted by the copyright holder.
// Graphica Software Pty. Ltd.
// 851 Brunswick St. North
// North Fitzroy 3068
// Victoria, Australia.
// Ph: + 61 3 9481 3747
// Fax: + 61 3 9481 3747
// Email: info@melb.graphica.com.au
// http://www.graphica.com.au
//
#ifndef WNIOINIT_H_
#define WNIOINIT_H_

#if defined(_STDIO_LOG_) || defined(_STDIO_CONSOLE_)

#include "initcnt.h"

class WinStdIOInit {
    friend class InitCounter<WinStdIOInit>;
private:
    static void Start();
    static void Stop();
};

INITCNT_HEADER(WinStdIOInit)

#endif

#endif
```

```
//
// File: wnioinit.cpp
//
// Contents: Automatically create a console for
//           stdin, stdout and stderr.
//           Based on the value of the following
//           compiler flags:
//           _STDIO_LOG_ - do init, send to log
//           _STDIO_CONSOLE_ - do init, send to console
//
// Programmer: John Hartley
//
// Version: 1.0
//
// Date: 24/4/97
//
// License:
// Copyright (c) 1997 Graphica Software Pty. Ltd.
// This software is part of the "ObjectBlocks" (TM) reusable components library.
// Its use is allowed only if this notice is included intact and
// on the basis of permission being granted by the copyright holder.
// Graphica Software Pty. Ltd.
// 851 Brunswick St. North
// North Fitzroy 3068
// Victoria, Australia.
```

```
// Ph: + 61 3 9481 3747
// Fax: + 61 3 9481 3747
// Email: info@melb.graphica.com.au
// http://www.graphica.com.au
//

#include "wnioinit.h"
#include "winstdio.h"

#if defined(_STDIO_CONSOLE_)

INITCNT_DEFINE(WinStdIOInit)

// #if defined(_STDIO_LOG_)
// #elif defined(_STDIO_CONSOLE_)
// #endif

void WinStdIOInit::Start()
{
    StdIoConsole(1);
}

void WinStdIOInit::Stop()
{
    StdIoConsole(0);
}

#endif
```

This source code is part of the “ObjectBlocks” reusable software library which is a work in progress of pattern based classes and utilities used to support Graphica Software Pty. Ltd. software development projects.

**End of Document**