Coding Cookbook

By

Nicholas Kingsley

ISBN : 978-1-4709-5717-9

Introduction

This book contains a fair few of my routines that I have written over the many years of programming on a PC, starting from DarkBasic Professional and C and then onto the wonderful GLBasic.

Unlike most other programming cookbooks, this one will, where possible describe how the routine came about and what it was originally used for.

There aren't any details on how to use the examples and you wouldn't, in most cases, be able to use the code as set out as they usually need other routines, but the code listed here should help you in your own programming endeavours and could give you a few ideas.

Formatting has been left as it was in the original file.

Code is listed for the following languages :

GLBasic
C
Javascript
DarkBasic Professional
DarkGame SDK
BlitzMax
PureBasic

and one bit of PHP code

Index

GLBasic

BufferReading

This routine was devised to allow large blocks of data to be read in quickly. This code was never really used, and was only written to see how easy it would be to access the usual C file I/O functions.

```
INLINE
       typedef unsigned int size_t;

       extern "C" int open(char * filename, int flags, int mode);
       extern "C" int close(int fd);
       extern "C" void * memset ( void * ptr, int value, size_t num );
       extern "C" int READ(int fd, void * ptr, int numbytes);

       #define      _O_RDONLY    0
       #define _O_WRONLY   1
       #define _O_RDWR           2

       #define _O_ACCMODE  (O_RDONLY|O_WRONLY|O_RDWR)

       #define      _O_APPEND    0x0008
       #define      _O_CREAT     0x0100
       #define      _O_TRUNC     0x0200
       #define      _O_EXCL             0x0400

       #define      _O_TEXT             0x4000
       #define      _O_BINARY    0x8000
       #define      _O_RAW       _O_BINARY

       #define      _O_TEMPORARY 0x0040

       #define _O_RANDOM   0x0010
       #define _O_SEQUENTIAL     _O_RANDOM
       #define _O_SHORT_LIVED    0x1000
ENDINLINE

FUNCTION BufferOpen%:fileName$
INLINE
       RETURN (DGNat) open((char *) fileName_Str.c_str(),_O_RDONLY | _O_BINARY,0);
ENDINLINE
ENDFUNCTION

FUNCTION BufferClose%:handle%
INLINE
       RETURN (DGNat) close(handle);
ENDINLINE
ENDFUNCTION

FUNCTION BufferRead$:handle%,BYREF result%
LOCAL temp$

       temp$=""

INLINE
       char buffer[4097];
       int res;

       memset(&buffer,(char) 0,sizeof(buffer));
       res=read(handle,(char *) &buffer,sizeof(buffer)-1);
       result=res;
       temp_Str.assign((char *) &buffer);
ENDINLINE

       RETURN temp$
ENDFUNCTION
```

C Headers

This set of “extern” function definitions allows various C routines to be called. This was created as, to start with, all of my C routines had different sets of function definitions.

```
INLINE
typedef int size_t;

// C commands
// mem... commands
extern "C" void *memset(void *s, int c, size_t n);
extern "C" void *memcpy(void *dest, const void *src, size_t n);
extern "C" int memcmp(const void *s1, const void *s2, size_t n);
extern "C" void * memmove(void * destination, const void * source, size_t num );

// Memory allocation/freeing commands
extern "C" void *realloc( void * ptr, size_t size );
extern "C" void *malloc ( size_t size );
extern "C" void *calloc(size_t nmemb, size_t size);
extern "C" void free(void *ptr);

// String commands
extern "C" size_t strlen(const char *s);
extern "C" char * strcpy ( char * destination, const char * source );
extern "C" char * strncpy ( char * destination, const char * source, size_t num );
extern "C" int strcmp ( const char * str1, const char * str2 );
extern "C" int sprintf ( char * str, const char * format, ... );

// Low level I/O
extern "C" int access(const char *pathname, int mode);
extern "C" int open(const char *pathname, int flags);
extern "C" size_t READ(int fd, void *buf, size_t count);
extern "C" int write(int fd, char *buf, size_t count);
extern "C" int close(int fd);
extern "C" int rename(const char *_old, const char *_new);

// These are used with the open function
#ifdef WIN32
      typedef struct __RECT {
            long left;
            long top;
            long right;
            long bottom;
            } __RECT;

      typedef int HDC;

      /* Specifiy one of these flags TO define the access mode. */
      #define      _O_RDONLY    0
      #define _O_WRONLY   1
      #define _O_RDWR           2

      /* Mask FOR access mode bits IN the _open flags. */
      #define _O_ACCMODE  (_O_RDONLY|_O_WRONLY|_O_RDWR)

      #define      _O_APPEND    0x0008 /* Writes will add TO the END of the file. */

      #define      _O_RANDOM    0x0010
      #define      _O_SEQUENTIAL0x0020
      #define      _O_TEMPORARY 0x0040 /* Make the file dissappear after closing.
                                 * WARNING: Even IF NOT created by _open! */
      #define      _O_NOINHERIT 0x0080

      #define      _O_CREAT     0x0100 /* Create the file IF it does NOT exist. */
      #define      _O_TRUNC     0x0200 /* Truncate the file IF it does exist. */
      #define      _O_EXCL            0x0400 /* Open only IF the file does NOT exist. */

      #define _O_SHORT_LIVED  0x1000

      /* NOTE: Text is the DEFAULT even IF the given _O_TEXT bit is NOT on. */
      #define      _O_TEXT            0x4000 /* CR-LF IN file becomes LF IN memory. */
      #define      _O_BINARY    0x8000 /* INPUT AND output is NOT translated. */
      #define      _O_RAW       _O_BINARY
```

```
	#ifndef	_NO_OLDNAMES

	/* POSIX/Non-ANSI names FOR increased portability */
	#define	O_RDONLY	_O_RDONLY
	#define O_WRONLY	_O_WRONLY
	#define O_RDWR		_O_RDWR
	#define O_ACCMODE	_O_ACCMODE
	#define	O_APPEND	_O_APPEND
	#define	O_CREAT		_O_CREAT
	#define	O_TRUNC		_O_TRUNC
	#define	O_EXCL		_O_EXCL
	#define	O_TEXT		_O_TEXT
	#define	O_BINARY	_O_BINARY
	#define	O_TEMPORARY	_O_TEMPORARY
	#define O_NOINHERIT	_O_NOINHERIT
	#define O_SEQUENTIAL	_O_SEQUENTIAL
	#define	O_RANDOM	_O_RANDOM

	#define SM_CXSCREEN 0
	#define SM_CYSCREEN 1
	#define HWND_BOTTOM	((HWND)1)
	#define HWND_NOTOPMOST	((HWND)(-2))
	#define HWND_TOP	((HWND)0)
	#define HWND_TOPMOST	((HWND)(-1))
	#define HWND_DESKTOP	(HWND)0
	#define HWND_MESSAGE	((HWND)(-3))

	#define SWP_DRAWFRAME		0x0020
	#define SWP_FRAMECHANGED	0x0020
	#define SWP_HIDEWINDOW		0x0080
	#define SWP_NOACTIVATE		0x0010
	#define SWP_NOCOPYBITS		0x0100
	#define SWP_NOMOVE		0x0002
	#define SWP_NOSIZE		0x0001
	#define SWP_NOREDRAW		0x0008
	#define SWP_NOZORDER		0x0004
	#define SWP_SHOWWINDOW		0x0040
	#define SWP_NOOWNERZORDER	0x0200
	#define SWP_NOREPOSITION	SWP_NOOWNERZORDER
	#define SWP_NOSENDCHANGING	0x0400
	#define SWP_DEFERERASE		0x2000
	#define SWP_ASYNCWINDOWPOS	0x4000

	#define SW_HIDE		0
	#define SW_SHOWNORMAL	1
	#define SW_SHOWNOACTIVATE	4
	#define SW_SHOW		5
	#define SW_MINIMIZE	6
	#define SW_SHOWNA	8
	#define SW_SHOWMAXIMIZED	11
	#define SW_MAXIMIZE		12
	#define SW_RESTORE		13
	#define HORZRES	8
	#define VERTRES	10

	extern "C" __stdcall int GetSystemMetrics(int);
	extern "C" __stdcall int GetWindowRect(HWND hWnd,struct __RECT *lpRect);
	extern "C" __stdcall int GetClientRect(HWND hWnd,struct __RECT *lpRect);
	extern "C" __stdcall int SetWindowTextA(HWND hWnd,const char *lpString);
	extern "C" __stdcall HWND GetDesktopWindow(void);
	extern "C" __stdcall int SetWindowPos(HWND hWnd,HWND hWndInsertAfter,int X,int
Y,int cx,int cy,int uFlags);
	extern "C" __stdcall int EnumDisplaySettingsA(const char*, unsigned int, void*);
	extern "C" __stdcall HWND GetForegroundWindow(void);
	extern "C" __stdcall int GetLastError(void);
	extern "C" __stdcall int GetSystemMetrics(int nIndex);
	extern "C" __stdcall HDC GetDC(HWND);
	extern "C" __stdcall int GetDeviceCaps(HDC,int);

	#endif
#elif _WIN32_CE
	/*
	 * Flag values FOR open(2) AND fcntl(2)
	 * The kernel adds 1 TO the open modes TO turn it into some
	 * combination of FREAD AND FWRITE.
	 */
	#define	_FOPEN	(-1)	/* from sys/file.h, kernel use only */
```

```
        #define      _FREAD       0x0001 /* READ enabled */
        #define      _FWRITE             0x0002 /* write enabled */
        #define      _FAPPEND     0x0008 /* append (writes guaranteed at the END) */
        #define      _FMARK       0x0010 /* internal; mark during gc() */
        #define      _FDEFER             0x0020 /* internal; defer FOR NEXT gc pass */
        #define      _FASYNC             0x0040 /* signal pgrp when DATA ready */
        #define      _FSHLOCK     0x0080 /* BSD flock() shared lock present */
        #define      _FEXLOCK     0x0100 /* BSD flock() exclusive lock present */
        #define      _FCREAT             0x0200 /* open with file create */
        #define      _FTRUNC             0x0400 /* open with truncation */
        #define      _FEXCL       0x0800 /* error on open IF file exists */
        #define      _FNBIO       0x1000 /* non blocking I/O (sys5 style) */
        #define      _FSYNC       0x2000 /* do all writes synchronously */
        #define      _FNONBLOCK   0x4000 /* non blocking I/O (POSIX style) */
        #define      _FNDELAY     _FNONBLOCK   /* non blocking I/O (4.2 style) */
        #define      _FNOCTTY     0x8000 /* don't assign a ctty on this open */

        #define      O_ACCMODE    (O_RDONLY|O_WRONLY|O_RDWR)

        #define      O_RDONLY     0            /* +1 == FREAD */
        #define      O_WRONLY     1            /* +1 == FWRITE */
        #define      O_RDWR       2            /* +1 == FREAD|FWRITE */
        #define      O_APPEND     _FAPPEND
        #define      O_CREAT             _FCREAT
        #define      O_TRUNC             _FTRUNC
        #define      O_EXCL       _FEXCL
        /*     O_SYNC       _FSYNC       NOT posix, defined below */
        /*     O_NDELAY     _FNDELAY     set IN include/fcntl.h */
        /*     O_NDELAY     _FNBIO              set IN 5include/fcntl.h */
        #define      O_NONBLOCK   _FNONBLOCK
        #define      O_NOCTTY     _FNOCTTY
        /* FOR machines which care - */
        #if defined (_WIN32) || defined (__CYGWIN__)
        #define _FBINARY        0x10000
        #define _FTEXT          0x20000
        #define _FNOINHERIT 0x40000

        #define O_BINARY    _FBINARY
        #define O_TEXT              _FTEXT
        #define O_NOINHERIT _FNOINHERIT

        /* The windows header files define versions with a leading underscore.  */
        #define _O_RDONLY   O_RDONLY
        #define _O_WRONLY   O_WRONLY
        #define _O_RDWR             O_RDWR
        #define _O_APPEND   O_APPEND
        #define _O_CREAT    O_CREAT
        #define _O_TRUNC    O_TRUNC
        #define _O_EXCL             O_EXCL
        #define _O_TEXT             O_TEXT
        #define _O_BINARY   O_BINARY
        #define _O_RAW              O_BINARY
        #define _O_NOINHERIT        O_NOINHERIT
        #endif

        #ifndef      _POSIX_SOURCE

        #define      O_SYNC       _FSYNC

        #endif
#else
        // For all other platforms
        #define _O_ACCMODE       00003
        #define _O_RDONLY           00
        #define _O_WRONLY           01
        #define _O_RDWR             02
        #define _O_CREAT         00100   /* NOT fcntl */
        #define _O_EXCL          00200   /* NOT fcntl */
        #define _O_NOCTTY        00400   /* NOT fcntl */
        #define _O_TRUNC         01000   /* NOT fcntl */
        #define _O_APPEND        02000
        #define _O_NONBLOCK      04000   /* NOT fcntl */
        #define _O_NDELAY        O_NONBLOCK
#endif

#ifndef IPHONE
        // This is for non-windows platforms
```

```
      extern "C" __stdcall void SDL_WM_SetCaption(const char *,const char *);
#endif
ENDINLINE
```

CRC32

This routine was created before GLBasic was equipped with the ENCRYPT/DECRYPT set of commands, and is used to generate a CRC 32 value for a set of data. This code was originally used when I was working at Plastics Software (and was used, if I remember correctly as part of the copy protection system). It required no modification to be used in GLBasic.

```
//! This calculates a CRC-32 value for a file
//\param fileName$ - This is the filename that you want to calculate a CRC value for
//\return - A signed 32-bit CRC value
FUNCTION calculateCRC32%:fileName$
LOCAL channel%      =      1
LOCAL bufferSize%   =      1024
LOCAL d%[];  DIM d%[bufferSize%]
LOCAL amount%
LOCAL crc%
LOCAL loop%
LOCAL length

       crc%=0
       IF DOESFILEEXIST(fileName$)=TRUE
              length=GETFILESIZE(fileName$)
              IF OPENFILE(channel%,fileName$,1)
                     DEBUG "length : "+length+"\n"

                     // Read in bytes
                     FILESEEK channel%,0,0
                     amount%=readInBytes(channel%,d%[],bufferSize%,length)
                     WHILE amount%>0
                            FOR loop%=0 TO amount%-1
                                   crc%=calcCRC32(d%[loop%],crc%)
                            NEXT

                            amount%=readInBytes(channel%,d%[],bufferSize%,length)
                            DEBUG "."
                     WEND

                     crc%=calcCRC32End(crc%)
                     CLOSEFILE channel%
              ELSE
                     DEBUG "Error : "+GETLASTERROR$()+"\n"
              ENDIF
       ENDIF

       RETURN crc%
ENDFUNCTION

@FUNCTION readInBytes%:handle%,d%[],size%,length
LOCAL loop%
LOCAL pos
LOCAL diff

       loop%=0
       WHILE ENDOFFILE(handle%)=FALSE AND loop%<size%
              pos=FILEPOSITION(handle%)
              diff=length-pos
              SELECT diff
                     CASE   1
                                          READUBYTE handle%,d%[loop%]
                     CASE   2
                                          READUWORD handle%,d%[loop%]
                     CASE   3
                                          READUBYTE handle%,d%[loop%]
                                          INC loop%,1
                                          IF loop%<size% AND ENDOFFILE(handle%)=FALSE THEN
READUWORD handle%,d%[loop%]
                     CASE   >3
                                          READULONG handle%,d%[loop%]
              ENDSELECT
              INC loop%,1
       WEND

       RETURN loop%
ENDFUNCTION
```

```
INLINE
const unsigned long CRC[]={
              0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706af48f,
              0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
              0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91, 0x1db71064, 0x6ab020f2,
              0xf3b97148, 0x84be41de, 0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
              0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f, 0x63066cd9,
              0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
              0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b, 0x35b5a8fa, 0x42b2986c,
              0xdbbbc9d6, 0xacbcf940, 0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
              0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116, 0x21b4f4b5, 0x56b3c423,
              0xcfba9599, 0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
              0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190, 0x01db7106,
              0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
              0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818, 0x7f6a0dbb, 0x086d3d2d,
              0x91646c97, 0xe6635c01, 0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
              0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6, 0x12b7e950,
              0x8bbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
              0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2, 0x4adfa541, 0x3dd895d7,
              0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
              0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c, 0x270241aa,
              0xbe0b1010, 0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
              0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17, 0x2eb40d81,
              0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
              0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683, 0xe3630b12, 0x94643b84,
              0x0d6d6a3e, 0x7a6a5aa8, 0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
              0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d, 0x806567cb,
              0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
              0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5, 0xd6d6a3e8, 0xa1d1937e,
              0x38d8c2c4, 0x4fdff252, 0xd1bb67f1, 0xa6bc5767, 0x3fb506dd, 0x48b2364b,
              0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60, 0xdf60efc3, 0xa867df55,
              0x316e8eef, 0x4669be79, 0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
              0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f, 0xc5ba3bbe, 0xb2bd0b28,
              0x2bb45a92, 0x5cb36a04, 0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b, 0x5bdeae1d,
              0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a, 0x9c0906a9, 0xeb0e363f,
              0x72076785, 0x05005713, 0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38,
              0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21, 0x86d3d2d4, 0xf1d4e242,
              0x68ddb3f8, 0x1fda836e, 0x81be16cd, 0xf6b9265b, 0x6fb077e1, 0x18b74777,
              0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c, 0x8f659eff, 0xf862ae69,
              0x616bffd3, 0x166ccf45, 0xa00ae278, 0xd70dd2ee, 0x4e048354, 0x3903b3c2,
              0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db, 0xaed16a4a, 0xd9d65adc,
              0x40df0b66, 0x37d83bf0, 0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f, 0x30b5ffe9,
              0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6, 0xbad03605, 0xcdd70693,
              0x54de5729, 0x23d967bf, 0xb3667a2e, 0xc4614ab8, 0x5d681b02, 0x2a6f2b94,
              0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d
              };
ENDINLINE

@FUNCTION calcCRC32%:value%,crc%
INLINE
              RETURN (CRC[(unsigned char) (crc^(unsigned long) value)]^((crc<<8) &
0x00FFFFFF));
ENDINLINE
ENDFUNCTION

@FUNCTION calcCRC32End%:crc%
INLINE
       RETURN (crc^0xFFFFFFFF);
ENDINLINE
ENDFUNCTION
```

Hosting Network Game

This routine was designed to allow the user to create or host a network game using a lobby system so that host players can see all players and all clients can join without needing to know the host's IP address.

It more or less works, but always seemed to be very inefficient. This was created, originally for one of my games, to see if I could get an efficient and easy to use network play system up and running.

```
// -------------------------------- //
// Project: TestHostJoinNetworkGame
// Start: Sunday, January 25, 2009
// IDE Version: 6.143

// Network messages
// This is a broadcast message
// A broadcast message will contain the following information
// 2 character program ID (in hex), consisting of :
// t12vsjrr rrrrrrrr iiiiiiii iiiiiiii
// t - If this is 1, then its a test program
// 1 - If this is 1, then the program is a timed demo program
// 2 - If this is 1, then the program is a non-timed demo program
// v - If this is 1, then the exact program version is needed to connect
// s - If this is 1, then the server is allowed to play on its own (single player mode
or sandbox mode)
// r - Reserved for future expansion
// i - This is the program ID

GLOBAL SEPERATOR$                           =      ","
GLOBAL NETMESSAGE_BROADCAST$           =      "BDC"
GLOBAL BROADCAST_TESTPROGRAM%          =      2147483648   // 31
GLOBAL BROADCAST_TIMEDDEMO%                 =      1073741824   // 30
GLOBAL BROADCAST_NONTIMEDDEMO%         =      536870912    // 29
GLOBAL BROADCAST_EXACTVERSION%         =      268435456    // 28
GLOBAL BROADCAST_SINGLEMODE%           =      134217728    // 27
GLOBAL BROADCAST_JOINATANYTIME%        =      67108864     // 26

GLOBAL NETMESSAGE_HOSTDISCONNECT$      =      "HDC"  // Host is disconnecting
GLOBAL NETMESSAGE_CLIENTDISCONNECT$    =      "CDC"  // Client is disconnecting
GLOBAL NETMESSAGE_HOSTKICK$                 =      "KCK"  // Host wants to kick a client
GLOBAL NETMESSAGE_REQUESTTOJOIN$ =     "RTJ"  // Client wants to join
GLOBAL NETMESSAGE_NOROOMLEFT$          =      "NRL"  // No more clients can join
GLOBAL NETMESSAGE_ALLOWJOIN$           =      "ALJ"  // Server has allowed join
GLOBAL NETMESSAGE_CLIENTREADY$         =      "CIR"  // Client is ready
GLOBAL NETMESSAGE_REQUESTPRESENCE$     =      "RFP"  // A request for presence has been
made
GLOBAL NETMESSAGE_PRESENCERESPONSE$    =      "PRE"  // The response to a request for
presence
GLOBAL NETMESSAGE_MESSAGE$             =      "MSG"  // All messages are sent to everyone
GLOBAL NETMESSAGE_SENDPLAYERINFO$      =      "SPP"  // Send player information to all
clients
GLOBAL NETMESSAGE_GAMEINPROGRESS$      =      "GIP"  // A Gane is in progress
GLOBAL NETMESSAGE_GAMEDATA$                 =      "GAD"  // Game data for other
machines to update display

// This is for detecting whether a server (or client) has disconnected
// If there is no reply within the allotted time, then its regarded as no longer being
present
TYPE tACK
       timeForACK
       timeForACKResponse
ENDTYPE

// Client and server types
// This is a list of client machines connected
TYPE tClientList
       ignore%                         // If this is TRUE, then the client is ignored
because it no longer exists
       isReady%
       ipAddress%
       clientName$
       ack AS tACK
ENDTYPE

// This is a list of availiable servers
```

```
TYPE tServerList
	connected%		// Is the player connected to this server ?
	ipAddress%		// IP address of server
	programID%		// Details of program being hosted
	version%		// Version of program
	numPlaying%		// Current number of people connected TO the server
	maxPlayers%		// Maximum number of people allowed
	operatingSystem$	// Operating system
	hostName$
	ack AS tACK
ENDTYPE

// This contains all players, excluding the connecting computer


GLOBAL clientList[] AS tClientList
GLOBAL serverList[] AS tServerList
// GLOBAL listOfPlayers[] AS tListOfPlayers

GLOBAL listKey$="listKey"
GLOBAL messageListKey$="messageListKey"

GLOBAL NOT_SELECTED%		=	-1
GLOBAL NOT_FOUND%		=	-1

GLOBAL TIMEFORACK		=	1000.0
GLOBAL RESPONSETIMEFORACK =	2500.0

GLOBAL STATE_CONNECTIONSETUP%	=	0
GLOBAL STATE_INITIALISE%	=	1
GLOBAL STATE_SYNCPLAYERS%	=	2
GLOBAL STATE_SETUPGAME%			=	3
GLOBAL STATE_SETUPLEVEL%	=	4
GLOBAL STATE_DOGAME%			=	5

FUNCTION hostJoinNetworkGame:isHost%,maxPlayers%=4,maxNameLength%=8
LOCAL playerName$

	DIM clientList[0]
	DIM serverList[0]   // Does this for clients, as they can list many servers

	maxPlayers%=bAND(maxPlayers%,255)

	IF isHost%=TRUE
		playerName$="Host Plyr"
	ELSE
		playerName$="Join Plyr"
	ENDIF

	IF changeName(-1,playerName$,maxNameLength%)=FALSE
		SYSTEMPOINTER FALSE
		RETURN FALSE
	ENDIF

	RETURN setupDisplayNetwork(isHost%,playerName$,maxPlayers%,BROADCAST_TESTPROGRAM%
+BROADCAST_SINGLEMODE%+1)
ENDFUNCTION

// Get player to enter their name
FUNCTION changeName%:playerID%,BYREF name$,maxNameLength%
LOCAL nameKey$="nameKey"
LOCAL okKey$="okKey"
LOCAL cancelKey$="cancelKey"
LOCAL titleText$="Enter Your Name"
LOCAL okText$="OK"
LOCAL cancelText$="CANCEL"
LOCAL maxWindowWidth%
LOCAL maxWindowHeight%
LOCAL screenWidth%
LOCAL screenHeight%
LOCAL fontWidth%
LOCAL fontHeight%
LOCAL temp$

	GETSCREENSIZE screenWidth%,screenHeight%
	GETFONTSIZE fontWidth%,fontHeight%
```

```
	maxWindowWidth%=MAX(maxNameLength%+1,LEN(titleText$))+2                    // Title
length.  Now take into account the
	INC maxWindowWidth%,LEN(okText$)+LEN(cancelText$)+2                        //
OK/Cancel button, ...
	maxWindowWidth%=maxWindowWidth%*fontWidth%                                       //
And multiply by the font width

	maxWindowHeight%=(fontHeight%*2)+2

	DDgui_UpdateFont(TRUE)
	DDgui_pushdialog((screenWidth%-maxWindowWidth%)/2,(screenHeight%-maxWindowHeight
%)/2,maxWindowWidth%,maxWindowHeight%)
	DDgui_set("","MOVEABLE", TRUE) // can move the dialog at top bar
	DDgui_set("","TEXT",titleText$)
	DDgui_text(nameKey$,name$,MAX(maxNameLength%+1,LEN(titleText$))*fontWidth
%,fontHeight%+4)
	DDgui_set(nameKey$, "TEXT",MID$(DDgui_get$(nameKey$, "TEXT"),0,maxNameLength%))

	DDgui_button(okKey$,okText$,(LEN(okText$)*fontWidth%)+fontWidth%,fontHeight%+4)
	DDgui_button(cancelKey$,cancelText$,(LEN(cancelText$)*fontWidth%)+fontWidth
%,fontHeight%+4)

	WHILE TRUE
		ALPHAMODE 0.0
		DDgui_show(FALSE) // show the dialog + handle widgets

		IF DDgui_get(okKey$,"CLICKED")
			// Get the name out of the dialog
			temp$=MID$(DDgui_getitemtext$(nameKey$,0),0,maxNameLength%)
			DDgui_set(nameKey$,"TEXT",temp$)

			IF temp$<>""
				name$=temp$
				DDgui_popdialog()
				SYSTEMPOINTER FALSE
				RETURN TRUE
			ELSE
				DDgui_msg("Please enter a name",FALSE)
			ENDIF
		ENDIF

		IF DDgui_get(cancelKey$,"CLICKED")
			DDgui_popdialog()
			SYSTEMPOINTER FALSE
			RETURN FALSE
		ENDIF

		SHOWSCREEN
		HIBERNATE
	WEND
ENDFUNCTION

// This is the main screen used by host and clients
FUNCTION setupDisplayNetwork%:isHost%,playerName$,maxPlayers%,programID%
LOCAL backKey$="backKey"
LOCAL backText$="BACK"
LOCAL kickKey$="kickKey"
LOCAL kickText$="KICK"
LOCAL startKey$="startKey"
LOCAL startText$="START"
LOCAL joinKey$="joinKey"
LOCAL joinText$="JOIN"
LOCAL readyKey$="readyKey"
LOCAL readyText$="READY"
LOCAL disconnectKey$	=	"disconnectKey"
LOCAL disconnectText$	=	"DISCONNECT"

LOCAL messageKey$="messageKey"
LOCAL sendKey$="sendKey"

LOCAL numPlayers%
LOCAL port%=50130   // Apparently its best to use ports above 49152
LOCAL isJoined%
LOCAL socket%
LOCAL broadcastIP%
LOCAL timeToSendBroadcast
LOCAL result%
```

```
LOCAL msg$
LOCAL length%
LOCAL thisIP%                   // IP Address if this computer
LOCAL messageList$[]
LOCAL selLine%
LOCAL clientCount%        // Number of clients connected to a server
LOCAL connectedServer%
LOCAL ver%                      // Version string as a 32-bit integer
LOCAL cList AS tClientList
LOCAL allReady%
LOCAL screenWidth%
LOCAL screenHeight%
LOCAL width% =      640
LOCAL height%=      480
LOCAL speed
LOCAL gameStarted%
LOCAL state%
LOCAL index%
LOCAL thisIndex%

       IF SOCK_INIT()=FALSE
              DDgui_msg("Unable to initialise socket system",FALSE)
              RETURN FALSE
       ENDIF

       socket%=SOCK_UDPOPEN(port%)
       IF socket%<0
              DDgui_msg("Unable to open UDP port : "+NETGETLASTERROR$(),FALSE)
              RETURN FALSE
       ENDIF

       initAppTime()

       broadcastIP%=SOCK_GETIP("255.255.255.255")
       thisIP%=SOCK_GETIP("")

       numPlayers%=0
       isJoined%=FALSE
       timeToSendBroadcast=0.0
       selLine%=NOT_SELECTED%
       msg$=""
       clientCount%=0
       connectedServer%=NOT_SELECTED%
       ver%=versionToInt("0.0.0.1")
       msg$=""
       gameStarted%=FALSE
       state%=STATE_CONNECTIONSETUP%

       GETSCREENSIZE screenWidth%,screenHeight%

       IF isHost%=TRUE
              // Setup the host network
              createHostJoinWindow(listKey$,"",screenWidth%,screenHeight,width%,height%)
              DDgui_spacer(0,1)
              DDgui_button(backKey$,backText$,0,0)
              DDgui_button(kickKey$,kickText$,0,0)
              DDgui_button(startKey$,startText$,0,0)

              INC numPlayers%,1
              changeHostTitle(numPlayers%,maxPlayers%)
       ELSE
              // Setup the client network
              createHostJoinWindow(listKey$,"",screenWidth%,screenHeight,width%,height%)
              DDgui_button(joinKey$,joinText$,0,0)

              DDgui_button(readyKey$,readyText$,0,0)
              DDgui_button(backKey$,backText$,0,0)
              DDgui_button(disconnectKey$,disconnectText$,0,0)

              changeClientTitle(FALSE,"",0)
       ENDIF

       // Add in the message part
       DDgui_spacer(0,1)
       DDgui_list(messageListKey$, "", width%-16,48)
       DDgui_text(messageKey$,"",width%/2,0)
       DDgui_button(sendKey$,"Send",0,0)
```