FreeDoor Version 1.x

BBS Door Authoring Programming Library
For Virtual Pascal & Borland Pascal

Developed and programmed by Mike Hodgson
Documentation by Mike Hodgson and Sean Dennis

Revised: December 11th, 2001

Disclaimer

By compiling and using the software contained in this package, you avail Mike Hodgson of any responsibility for damage done to your computer hardware and/or software.

What is FreeDoor?

FreeDoor is a freeware door authoring programming library for Virtual Pascal and Borland Pascal to enable the door author to create doors in the operating systems that the 3 compilers support, namely Win32 and OS/2, and DOS at this time. FreeDoor also contains EleCom, an excellent opensource programming library that allows use of Internet functions (namely sockets and telnet functions) for programs.

How do I set up FreeDoor for use in my system?

Virtual Pascal Setup

Here's the recommended setup (note that in these examples we'll use C:\VP21 as our root directory):

1) Create a directory called FREEDOOR in your VP directory, such as **C:\VP21\FREEDOOR**.

2) Unzip FreeDoor (using the –D switch if you're using PKUNZIP) into the above-mentioned directory.

3) In Virtual Pascal, click on Options->Directories.

4) Under "Include Directories", put in: **C:\VP21\FREEDOOR\SOURCE;C:\VP21\FREEDOOR\ELECOM;**

5) Under "Unit Directories", put in: **C:\VP21\FREEDOOR\SOURCE;C:\VP21\FREEDOOR\ELECOM;**

6) Under "Library Directories", put in: **C:\VP21\FREEDOOR\SOURCE;C:\VP21\FREEDOOR\ELECOM;**

7) Under "Resource Directories", put in: **C:\VP21\FREEDOOR\ELECOM\RES\W32;C:\VP21\FREEDOOR\ELECOM\OS2;**

8) Open **C:\VP21\FREEDOOR\SOURCE\FREEDOOR.PAS**, click on Compile, make sure that the target platform is Win32 and click on Make (or press F9). Switch the target platform to OS2 and make that library.

Borland/Turbo Pascal Setup

1) Create a directory called FREEDOOR in your BP directory, such as **C:\BP7\FREEDOOR**.

2) Unzip FreeDoor (using the –D switch if you're using PKUNZIP) into the above-mentioned directory.

3) In Virtual Pascal, click on Options->Directories.

4) Under "Include Directories", put in: **C:\BP7\FREEDOOR\SOURCE;C:\BP7\FREEDOOR\ELECOM;**

5) Under "Unit Directories", put in: **C:\BP7\FREEDOOR\SOURCE;C:\BP7\FREEDOOR\ELECOM;**

6) Under "Library Directories", put in: **C:\BP7\FREEDOOR\SOURCE;C:\BP7\FREEDOOR\ELECOM;**

7) Open **C:\BP7\FREEDOOR\SOURCE\FREEDOOR.PAS**, click on Compile, and select Build. Make sure the platform is Real Mode. Protected mode is currently un-tested, and 16bit Windows is not supported.

How do I call FreeDoor from within my program?

That's easy too. Just include FreeDoor in your Uses clause, like:

```
Uses CRT, FreeDoor;
```

… and at the beginning of your main program, make sure you tell FreeDoor what the program's name is and call FreeDoor to initialize itself:

```
Program TestDoor;

Uses CRT, FreeDoor;

Begin
    ProgName := 'My First Door!';
    InitDoorDriver;

(…)
End.
```

What dropfiles does FreeDoor support?

At this time, FreeDoor will support **DORINFO1.DAT**, **DOOR.SYS** and **DOOR32.SYS** formats. There are no plans at this time for supporting other dropfile formats, but if there is demand for it, support will be considered.

How do I call my door from the command line?

The following command lines are supported in FreeDoor:

For local mode and non-BBS operation:
doorname.exe /L

For normal FOSSIL mode (# is port number):
doorname.exe /Dc:\path\to\door.sys /P#

For Telnet mode (# is port handle)
doorname.exe /Dc:\path\to\door.sys /T /P#

NOTE: If your system is using **DOOR32.SYS**, /T and /P# are not required.

Constants List

StatFore
A byte, can be used to change the foreground colour of the status bar. Accepts a value between $00 and $0F.

StatBack
A byte, can be used to change the background colour of the status bar. Accepts a value between $00 and $08.

Pause_String
Defines the text displayed at a pause prompt, by default displays <PAUSE>. Colour codes can be used.

ProgName
String containing your program's name. By default it is "Another FreeDoor Program". This should be set before calling InitDoorDriver.

fdInfo
Record containing information about the current user. The following is available: RealName, Handle, CityState, ACS, TimeLeft, TotalTime, ComPort, ConnType, Baud, Node, GraphMode, DropFile, DropType BBSID, RecPos. All are documented in **freedoor.inc**.

MASK_COLOUR
ANSI string representing foreground and background colour to be used in calls to CmaskInput.

Procedures / Functions List

InitDoorDriver : Boolean;

Initializes the door driver (duh!). This should be the first procedure used. Any FreeDoor constants should be changed before calling this.

DeInitDoorDriver;

Should always be called at the end of your program, or in the ExitProc.

CClrScr;

Clears the display. Cursor is located at 1,1.

CCLREol;

Clears to the end of the line. Cursor is located at the beginning of the current row.

CursorSave;

Saves cursor position. Used in conjunction with CursorRestore.

CursorRestore;

Restores cursor position saved by CursorSave.

CursorUp (Distance : Integer);

Move cursor up Distance number of lines.

CursorDown (Distance : Integer);

Move cursor down Distance number of lines.

CursorBack (Distance : Integer);

Move cursor left Distance number of spaces.

CursorForward (Distance : Integer);

Move cursor right Distance number of spaces.

CgotoXY (X,Y : Integer);

Locate cursor to position X,Y.

CCEnter(S : String);
Center a piece of text, S, on the screen.

ErrorWriteLn (S : String);
Shouldn't really be used by door author, but may need to be in some cases.
Doesn't update statusbar and time like CWriteLn does…

CWriteLn (S : String);

Display string S at the current location.

CWrite (S : String);

Same as CWriteLn but does not append trailing carriage return / line feed.

CGetChar (var Ch : Char);

Gets a character from local and/or remote keyboard and saves it in Ch.

CReadLn (var S : String);

Gets a string from local and/or remote keyboard and saves it in S.

CWriteLnLong (I : LongInt);

Writes a numeric variable (Byte, Integer, LongInt) to the display.

CWriteLong (I : LongInt);

Same as above but without trailing carriage return / line feed.

CGetByte (var B : Byte);

Get a single byte from the local or remote keyboard.

CReadLnLong (var L : LongInt);

Get a numeric value from the local or remote keyboard.

CPause;

Display the pause prompt and wait for a key press.

CWriteFile (FN : String);

Write a file to the local and remote screens.

CMaskInput (Mask : String; StrLength : Byte) : String;

Creates a mask input line. See Freedoor.inc for different mask strings. Returns the user's input.

CMaskInputPW (Mask : String; StrLength : Byte) : String;

Like CMaskInput, but echoes a * (asterisk) to the screen.

CCenter (S : String);

Center the string S on the screen.

CRight (S : String);

Right-align the string S on the screen.

CWindow (X1,Y1,X2,Y2 : Byte);

Create a window using the current color attributes from top left corner X1,Y1 to bottom right corner X2,Y2.

CSendToNode (S : String; Node : String);

Sends a message (S) to Node (Node). The message is given to the user on the next input prompt. Messages sent are automatically deleted if a user logs out. No checking is made to see wether a node is in use or not.

CGetFromNode;

Automatically called in input procedure. Displays any messages sent to the user and who they were sent by.

Acknowledgements and Contact

Microsoft Windows is © Microsoft, Inc.
OS/2 is © IBM Corp.
EleCom is © Maarten Bekers.
Win32crt is based on the FreePascal crt unit.

Thanks to the guys in #bbs on irc.lordlegacy.org for all the ideas and help.

Thanks to Sean Dennis for typing up these documents, I did some re-formatting, but pretty much all of the text is his ;o)

Some thanks go out to Rick Parrish (Manning, http://www.mannsoft.ca/) for the use of his input routines. Also to SWAG, I stole some of their code for releasing timeslices. And lastly to Michael Preslar for all the additions and fixes.

If you have any questions, comments, bugs or bug fixes, please do not hesitate to contact me by using one of the methods below.

E-Mail:	mike@mhdev.com
IRC:	look for coolio in #bbs on irc.lordlegacy.org
ICQ:	9434448
Web:	http://www.mhdev.com/