

# Chapter 22

## Blender Program

### 22.1 Development Tools

The blender program is written in the C language. The entire system is about 67,000 lines of code in about 200 files. With minor exceptions the system compiles and runs in the following environments:

Compiler	OS	Threading	Processor
Open Watcom	Windows	Win32	Intel x86
Visual C++ 2005 EE	Windows	POSIX	Intel x86
gcc	Linux	POSIX	Intel x86
gcc	Linux	POSIX	PowerPC (big endian)

The target platform is an Intel x86 processor running the Fedora Linux distribution.

The Open Watcom compiler is available from:

<http://www.openwatcom.com>

The Microsoft Visual C++ 2005 Express Edition compiler is available from:

<http://msdn.microsoft.com/vstudio/express/visualc>

Fedora Core Linux and the gcc compiler are available from:

<http://fedoraproject.org>

The system uses a third-party application called `ssl_proxy` to add encryption to the blender program's remote web function. `ssl_proxy` is available from:

<http://sourceforge.net/projects/sslproxy>

`ssl_proxy` does not run under Windows. The main difference between the blender program under Windows and the blender program under Linux is that under Linux all remote web access is securely encrypted. Under Windows remote web access is not encrypted and therefore not secure.

The blender system's documentation is prepared using the  $\text{\LaTeX}$  system.  $\text{\LaTeX}$  comes with Fedora Core Linux. A Windows version is available from:

<http://miktex.org>

The low-level functions to access the Automation Direct Terminator I/O racks were obtained from Host Engineering at:

<http://www.hosteng.com>

The Host Engineering source code was modified during development and the original code no longer works.

The database server runs the PostgreSQL database engine under Fedora Core Linux. PostgreSQL is included with the Fedora Core distribution. Additional information is available from:

<http://www.postgresql.org>

All of these development tools are free.

## 22.2 Batch Files

The following batch files are used to build the system:

Linux	Watcom	Microsoft	Function
<code>c</code>	<code>c.bat</code>	<code>mc.bat</code>	Compile something
<code>l</code>	<code>l.bat</code>	<code>ml.bat</code>	Link something
<code>-</code>	<code>r.bat</code>	<code>mr.bat</code>	Compile a resource
<code>build</code>	<code>build.bat</code>	<code>mbuild.bat</code>	Build the entire system
<code>blender</code>	<code>blender.exe</code>	<code>mblender.exe</code>	Program name

## 22.3 Architecture

The blender program is multithreaded, meaning it is composed of many smaller programs all running in concert. The program does not allocate or

free memory so it will not cause memory fragmentation. The program is not properly real-time because it must poll the Automation Direct Terminator I/O racks, which it does 10 times a second. The blender program runs at a higher priority than normal so that other programs like the `emacs` text editor do not slow it down. Reporting programs like the `gman` graph renderer are invoked at a lower priority than normal so they don't slow anything else down.

## 22.4 Performance

Under normal operation the blender program uses about 20% of the power of the 333MHz Pentium II CPU used for development. The CPU load on any modern computer would be negligible. The blender program uses about 7MB of RAM.

## 22.5 Dynamic IP Addresses

The blender program contains a built-in web server for remote Internet access. Internet servers usually have unchanging static IP addresses and use the standard DNS system to convert a server's name, like `farnham.kaytec.net`, to the server's IP number.

Some Internet connections only offer changeable dynamic IP addresses. The blender program has the facility to work with a dynamic IP address using another web server as an intermediary.

One of the programs of the blender system is a web server CGI program called `nexus`. The intermediate web server runs `nexus` in response to a particularly encoded web query sent by the blender program. This gives the intermediate server the IP address of the blender program and the port at which it will accept web traffic. The intermediate server's reply includes the IP address so the blender program knows it too.

The blender program queries the intermediate server this way every 20 seconds.

The standard DNS system is configured to convert the blender server's name, `farnham.kaytec.net` for example, into the intermediate server's static IP address.

When the intermediate server receives a plain HTTP query, like it would get from a web browser looking for `http://farnham.kaytec.net`, the intermediate server produces a page redirecting the user to a URL consisting of the blender program's dynamic IP address and port number, like `http://64.222.102.57:1120`.

Subsequent pages produced by the blender program use the numeric URL.

The intermediate server's redirection page shows the blender program's dynamic IP address. This can be used to log on to the database server and blender controller directly.

## 22.6 Source Code

Every `name.c` file has a corresponding `name.h` header file containing the function prototypes and symbols needed to use the `.c` file.

All the header files used by the system are included in `common.h`. Usually `common.h` is the only header file included by a `.c` file.

The blender system is composed of the following source code modules:

### 22.6.1 Non-forms

#### Database

<code>db.c</code>	Database functions
<code>dbtables.c</code>	Database table definitions
<code>dbutils.c</code>	Database utility functions
<code>libpq.c</code>	Linux database functions
<code>odbc.c</code>	Windows database functions

**Web Graphics and Reports**

graphics.c Web graphics form  
image.c Graphical drawing library  
imagedat.c Graphical data  
reports.c Web reports form

**Host Engineering**

hei.c High level I/O rack functions  
heii.c Low level network functions

**I/O rack interface**

iopconfigread.c Read iopoint.conf  
iopoint.c I/O rack manager thread

**Infrastructure**

event.c Event functions  
form.c Form processor  
mutex.c Mutex functions  
request.c Thread request functions  
thread.c Thread functions

**Other**

<code>font.c</code>	Font definitions
<code>const.c</code>	<code>const.conf</code> file functions
<code>crypt.c</code>	Encryption functions
<code>dce.c</code>	Time functions
<code>eventformat.c</code>	Event format function
<code>forms.c</code>	Form list
<code>fscalen.c</code>	<code>fscalen.c</code> template
<code>global.c</code>	Global variables
<code>kbdata.c</code>	Linux escape sequences
<code>linux.c</code>	Linux functions for Windows
<code>list.c</code>	List functions
<code>log.c</code>	Log file append
<code>machine.c</code>	State machine functions
<code>mainstop.c</code>	Send a stop request
<code>mem.c</code>	Memory management functions
<code>states.c</code>	State definitions
<code>timespan.c</code>	Utility parameter processor
<code>user.c</code>	User functions
<code>utils.c</code>	Utility functions
<code>web0.c</code>	Web server functions

**Threads**

<code>dbevent.c</code>	Database event logger
<code>dynip.c</code>	Dynamic IP handler
<code>kb.c</code>	Linux keyboard input
<code>main.c</code>	Main module
<code>sigterm.c</code>	Linux signal handler
<code>web1.c</code>	Web server listener
<code>web2.c</code>	Web server request handler

### Utility programs

<code>batchman.c</code>	Batch report generator
<code>dbman.c</code>	Database manager
<code>elist.c</code>	Event list generator
<code>forman.c</code>	Formula viewer
<code>gman.c</code>	Graphic generator
<code>kbtest.c</code>	Linux escape sequence viewer
<code>nexus.c</code>	Intermediate web site CGI program
<code>relay.c</code>	Serial relay controller
<code>rman.c</code>	Report generator
<code>slist.c</code>	Signal list generator
<code>touch.c</code>	Create a file
<code>userman.c</code>	Manage remote users

### Windows

<code>windata.c</code>	Windows-only data
<code>winmain.c</code>	Windows main thread

## 22.6.2 Forms

### Blending forms

<code>fadjust.c</code>	Blender parameter adjustment
<code>fconst.c</code>	Edit constants
<code>fmanual.c</code>	Manual control
<code>fselect.c</code>	Select formula for blending

### Formula and ingredient forms

<code>fformin.c</code>	Edit formula ingredient
<code>fformula.c</code>	Select formula ingredient for editing
<code>fformulas.c</code>	Select formula for editing
<code>fingred.c</code>	Edit ingredient
<code>fingreds.c</code>	Select ingredient for editing

**Low-level I/O forms**

facalib.c Analog input calibration  
fadconfig.c I/O rack configuration  
fainput.c Analog input viewer  
faoutput.c Analog output viewer  
fdinput.c Digital input viewer  
fdoutput.c Digital output viewer

**Other forms**

fexit.c Exit program  
fmain.c Main form

**Report forms**

fevents.c Event viewer  
fgraph.c Console analog input graphing  
fsession.c View remote user activity

**State machine forms**

falarm.c Alarm  
fbag.c Bag button and light  
fcooler.c Cooler  
fmixer.c Mixer  
fresbin.c Resin bin  
fscale1.c Scale 1  
fscale2.c Scale 2  
fscale3.c Scale 3  
fscale4.c Scale 4  
fscale5.c Scale 5  
fscreeener.c Screener  
fstate.c Master state machine

The `fscalen.c` files define  $n$ -related preprocessor symbols and then include `fscale.h`.