If the text runs off the edge of the page, use the Edit>Word Wrap command.

Contents of this file:

Introduction

The Red Baron 3D multiplayer server software was originally designed for operation of dedicated servers on high-bandwidth systems, and is provided here for advanced users. We encourage users to use and experiment with the server software, but be aware that doing so may require significant effort on your behalf. We've put enough information in this file to get you started. If you should experience problems, our tech support will be unable to provide you with any information beyond that you will find on this CD. If you do need additional information we would suggest visiting one of the Red Baron 3D related websites, such as our own REDBARONPLAYERS.COM site.

License information

The server software is subject to the license described in full in the file EULAServer.doc found in the server directory on the Red Baron 3D CD. Basically this license states that you are free to set up servers as long as these servers do not generate revenue through user fees, advertising, etc. If you wish to set up servers in any other manner, you should read the license documentation fully, and contact Sierra for written permission to do so.

How to set up a Red Baron II server:

1) Verify that you have the following files:
RB2SERVE.EXE
RB2SERVER.INI
RB2SERVER.SCS
LNDMRKS0.DAT
LNDMRKS1.DAT
LNDMRKS2.DAT
LNDMRKS3.DAT
LNDMRKS4.DAT
(probably COOKIE*.BIN files as well)

2) Copy these files to a directory on your machine, and change the properties of RB2SERVER.SCS and RB2SERVER.INI so they are not read-only.

3) Edit the RB2SERVER.INI file. This file controls the various game settings (game type, difficulty level, etc). Be sure that you change the description of your server so that there aren't lots of servers with the same desription!

4) Run RB2SERVE.EXE. You can double-click on it, or run it under a server monitor, batch file or other program scheduler (see additional documentation, if appropriate).

Running multiple Red Baron II servers on a single machine

1) Copy RB2SERVER.INI, and rename the copies so you have several INI files -- one for each copy of the server you want to be running.

2) Again, edit these INI files as appropriate for each game.

3) Now copy the RB2SERVER.SCS file for each INI file you created. We recommend that you name them similarly, to avoid future confusion. For example, if you have RB2SERVER_1.INI and RB2SERVER_2.INI, you should have RB2SERVER_1.SCS and RB2SERVER_2.SCS.

4) Edit each SCS file, and find the line that reads:
iniFile = ".\\rb2server.ini"
Change this line to reflect the name of the INI file appropriate for each game. Note that the double backslash is needed to separate any directories in this entry (e.g. C:\\redbaron\\server\\scs\\server1.scs")

5) Edit each SCS file, and find the line that reads:

```
    port = 47807;
```

Change this line to reflect the port numbers that you want each game to use. Note that each copy of the server must be using a different port number. If you are running the server behind a firewall, the port that you select must be opened to UDP message traffic.

6) If you have logging enabled (see below), you should change the logfile entries in each SCS file so that each server logs to a different set of files. If multiple SCS files point to the same log files, the first server started will have logging, but the rest will not log any information.

7) Run RB2SERVE.EXE, with the SCS file as a command line parameter. This will load the correct SCS file, and then the correct INI file.

What are those cookie*.bin files for, anyway?

The cookie*.bin files are used for security purposes. They tell the server which client executables can connect to it, so if a backward compatible patch is released, you can determine which of the versions are allowed on your server. At the bottom of the INI file is a section indicated by [Clients]. Entries in this section are of the form: Cookie=Cookie1002.bin. (Each file is named based on the internal version of the client executable. So you may have Cookie1002.bin, Cookie1003.bin and Cookie2001.bin). New cookie files will be made available with any patches that are released.

Registering with meta-servers

A meta-server is a server that keeps track of which game servers are running. If you want people to see your server, you have several options. First, you can give people your IP address directly, and they can look for your server specifically. Second, you can register your server with a Red Baron 3D meta-server. Anyone browsing for games within the game (with this meta-server in the client INI file) will be able to find your game. Third, WON.NET also provides some browsing capabilities through its web page. If you register with that meta-server, people looking for games through the web will be able to see your game. You can choose any or all of these options as appropriate for your server.

To register with a meta-server, look at the bottom of the INI file in a section indicated by [RegisterWith]. Entries in this section are of the form protocol:IPaddress:port. The protocol is either UDP (for Red Baron meta-servers), or TCP (for WON.NET meta-servers). The IPaddress can be either a numeric address, or an alias. Finally, the port number is the port that the meta-server is running on, NOT the port number that your server is using. Thus, the following entries are valid (these are just examples; meta-servers may or may not be running at these addresses):

```
    UDP:dyn-meta.dynamix.com:47801
    TCP:209.67.71.136:2667
```

Remote adminstration

Red Baron 3D servers can also act as Telnet clients to allow remote administration. This feature is disabled by default, for security reasons. If you would like to enable telnet to the server, edit the SCS file, and add an entry that reads:

```
    enableTelnet("password", port)
```

Obviously, you should change the password to be something less likely to be guessed. The port number can be the same as the port that the game is using, or it can be a different port number.

If this is set up, you can open a telnet session to the server, and act as though you were at the console of the machine running the server. You can dump/load the score database, eject players, etc.

NOTE There are security concerns involved with enabling Telnet on a machine connected to a network. Unless you are familiar with these issues, and have additional security measures in place, we recommend that you do not enable this feature.

Logging

There are various ways to log information on the server side. To control logging, edit the SCS file, and look for the entry that reads:

```
    openEventLog(100000, "rb2server")
```

Removing this line will disable server logging completely. The number is the total size of log files that will be generated. This size is typically split into several files, each of which is named based on the second parameter. In the case of the line above, the log files would be named rb2server0.log, rb2server1.log, etc.

The log file will always contain a header block for each time the server was started. Additionally, kill awards, chat messages and connects/disconnects can be selectively logged by setting the logKills, logChat and logConnects lines appropriately in the SCS file.

```
---------------------------------------------------------------
Score databases

The server maintains internal score information in a simple database format.  This database can be
manually loaded or saved out by the server administrator, or set to automatically load at startup and save
at shutdown.

To save the score database in an internal format from the server console, use the console command:
    dump_scores("filename.bin")
To save the score database from the server console in comma separated value (CSV) format, which can be
read by most spreadsheet programs and text editors, use the console command:
    csv_scores("filename.csv")
To manually load the score database in an internal format from the server console, use the command:
    load_scores("filename.bin")
Note that you cannot load the scores from the CSV file format; only the internal binary format can be read
back in.

Each of these settings is available automatically through the SCS file by adding or enabling the entry
that reads:
    scoreFileCSV="scoreFile.csv"
    scoreFileBIN="scoreFile.bin"
If either of these entries is found, the server will automatically save the score database in the
appropriate format when it shuts down.  By adding the entry that reads:
    scoreFileDaily="scoreFile.csv"
the server will save out the score database in CSV format at midnight of each day in the sim.  This would
allow periodic score updates for servers that are continually running.  Note that this update occurs at
midnight game time, not midnight real time in the server's location.
Finally, adding the entry:
    load_scores("scoreFile.bin")
to the SCS file will cause the server to load the specified score database at startup.  By setting both
this entry, and the scoreFileBIN entry, the same score database can be used by the server every time it is
restart.

---------------------------------------------------------------
Machine recommendations, and how many people can join

Obviously, the faster the server machine, the better.  However, the number of players that can join your
game is more limited by network bandwidth than server CPU.  We were able to run a dedicated server on a
Pentium 90 with 16Meg of RAM with no problems.  However, if you plan to run a server and a client on the
same machine, you should have AT LEAST a Pentium 200 to expect reasonable performance.

**NOTE**  If you are running a server and a client on the same machine, and the client is running in Glide
mode, you may need to minimize the server for the client to appear correctly.

If the server is on a dialup connection to the Internet, you can expect to reasonably run a 4-6 player
game with a 28.8k or 33.6k connection.  You may be able to get 8-10 players with a 56k connection.  For
the best performance, and for all large games, the server should be running on a dedicated machine with a
high-speed (i.e. T1 or better) connection.

The limit on the number of players in your game is controlled in the server INI file, with the
"maxPlayers" line.  If you set the maximum number of players higher than your network bandwidth will
reasonably support, those additional players will be allowed to join, but some players (not always the
newly joining players) will be dropped from the game and/or you will see increased lag.
```