Setting Up a Red Baron 3D Server Frequently Asked Questions

How do I setup a server to run over my IPX network?

Setting up an IPX based server is pretty simple and only requires a small change to one of the server's configuration files. Within the server folder locate the file named "rb2server.scs" and open it in a standard text editor like Notepad. Towards the top of the "rb2server.scs" file there is a configuration key labeled "protocol" and will look something like this:

protocol = "UDP";

Simply change the setting from UDP, which is the Internet connection type, to IPX. Also be sure that the "port" setting is a valid port number for the network connection type that you have chosen. Or set the "port" setting to 0, in which case, the Red Baron 3D server will select the next valid and available port for the specified network connection type. In most cases it easier to leave the port setting at 0 and let the server choose the network port that it will use. Save the changes and the next time you execute the server it will be running on an IPX network connection.

How can I setup up more than on server on the same machine?

The easiest way to run multiple servers on the same machine is to make copies of the server folder onto your hard drive naming them something like "Server 1", "Server 2", etc., or some other naming convention that helps you differentiate between the different servers. After you've done that you can modify all the separate "rb2server.ini" settings for the different servers to your desired game setup. The key to running multiple servers on the same machine is making sure that all the servers are running on different network port numbers than all the rest. If it doesn't matter which port number the server runs on, you can set the "port" setting in the "rb2server.scs" file to 0 for all the servers on your machine. In which case, when you run the server it will attempt to open the next valid, available port for that protocol. If you want to specify the physical port number for the server to run on, be sure to make each port number different for each server (i.e., "port = 47800;", "port = 47801;", "port = 47802;", etc.)

How can I setup a server from behind a firewall?

In order to run a server from behind a firewall, you have to tell the Red Baron 3D server to use a port that is currently open on that firewall for UDP message traffic. (*Note: if you don't know what ports may be available and/or open for UDP message traffic. Contact your system administrator about the specific details of your firewall and what ports may be open.)* Once you know what ports may be available for the server to run on. Open the "rb2server.scs" file and locate the setting "port =" and change it to reflect the port that is open on your firewall.

Example:

I know that port number 47800 is open for UDP traffic through my firewall. So I'll set my port setting to:

port = 47800;

How do I shut my server down when I'm done?

To shut your server down while it's running, type the command "quit()" in the server console window. This will cause your server to shutdown "gracefully" by broadcasting it's shutdown to any meta-servers that are running.

Can I run a server on the same machine as a Red Baron 3D client?

Yes, after the server is up and running you can simply run Red Baron 3D multiplayer and connect to your server. However, depending on the configuration of your machine (CPU speed, available RAM, etc.) the performance of the server may be affected with two processes running at the same time. (*Note: if you're running RB3D in glide mode, on some computer configurations, having a console application window open in the background will cause the glide screen to appear completely black within the simulation. The work-around for that is to minimize the server console window before starting up the Red Baron 3D multiplayer client and that will resolve the problem.)*

This console application window for the server is neat and all, but can I do anything in it?

From the console application window there are several commands that you, as the server administrator, can issue to the server:

- **quit()** – Shuts the server down
- **eject("player name")** – Will eject the specified player from the game. (*Note: this is case sensitive so make sure you have the case correct on all the letters in the player's name.)*
- **stats()** – Displays a player list and some other miscellaneous statistics about the server.
- **cls()** – Clears the console application window.
- **dump_scores("filename.bin")** – Saves the scores database in an internal format.
- **csv_scores("filename.csv")** – Saves the scores database in comma separated value (CSV) format, which can be read by most text editors.
- **load_scores("filename.bin")** – Manually loads the score database from an internal format. (*Note: You can only load scores from the internal binary file format.)*

How can I be sure all the players on my Server are using the same version of Red Baron?

For security purposes, the Red Baron server also comes with a series of files titled "Cookie*.bin;" These files tell the server which Red Baron clients can connect to it, so if a backward compatible patch is released you can determine which of the versions can join your server. This will then insure that all of the Red Baron clients on your server are using the same data and client versions. New cookie files will be made available with any patches that are released.

Towards the bottom of the "rb2server.ini" file is a section titled "[Clients]" this section stores the list of all the "cookie" files the server will use to negotiate it's version checking for any clients that join. The format for these entries are: "Cookie=CookieXXXX.bin" (*Note: the cookie filename is based off the version of the client executable.*)

If, for any reason, you want to disable you server's security checking towards the top of the "rb2server.ini" file is a setting labeled "securityEnabled" Setting this entry to 1 will enable your server's security checking, and setting this entry equal to 0 will disable all server security checking. (*Note: With this setting turned off, the server will perform no validation checking of the data any Red Baron 3D clients send to it. This means that the server will not prevent players from cheating, or using out of date or corrupted files. It is recommended that you leave this setting enabled whenever possible.*)

If I start a Red Baron server, how do I let other people see it?

There are actually several mechanisms for broadcasting the availability of your server to the world. The first being, you can simply give your IP address to the people you want to join your server, and they can have their Red Baron clients specifically look for your server. Or you can have your server broadcast it's availability with a Red Baron 3D meta-server. A meta-server is a server that lists game servers. Anyone

browsing for games within the game, with this meta-server in the client INI file, will be able to find your game. And lastly, WON.NET provides the ability to browse for games through its web page. If you register with WON.NET's meta-server, people looking for games from the web will be able to see and join your game.

I've specified a custom INI file in my "rb2server.scs" file, but my server keeps ejecting people.

There is the ability to specify a custom INI file name other than "rb2server.ini". However there is a catch. The Red Baron 3D server will still look for the "[Clients]" and the "[RegisterWith]" sections out of the "rb2server.ini" file. In no such file exists, the Red Baron server will fail to find the necessary information to broadcast it's availability to any of the Red Baron meta-servers. And all clients that join your server will fail their security validation checking, because the Red Baron server failed to load the necessary "cookie" files.

To resolve this issue you need to make sure there is an "rb2server.ini" file in the folder with the server as well as your custom INI file. Make sure that the "rb2server.ini" file contains both the "[Clients]" and "[RegisterWith]" sections and your server will function properly.

A Close Look at the "rb2server.ini" File

The following section is a brief overview of the settings in the Red Baron 3D server INI file, and what effect they have on your server. The setting descriptions in this section have been grouped together based off of their relation to one another, and may appear in a different order in the "rb2server.ini" file.

General Server Settings

Setting Name	Description	Setting Type
serverDesc	Name of your Red Baron 3D game server.	Text String
password	The server password if you choose to make your server password protected. The maximum length your password can be is 15 characters.	Text String
maxPlayers	Sets the maximum number of players that can be in the game at a given time.	Numeric
clientTimeoutSec	The length of time, in seconds, the server will wait before it times out a Red Baron 3D client.	Numeric
securityEnabled	If you want the server to use its security features to help prevent cheating, set this entry to 1. If you set this entry to 0, the server does no validation of data received from players with in the game. (***With this setting set to 0 the server will not do any checking to prevent cheating or invalid data packet processing and could result in very unexpected server behavior. Use at your own risk!***)	Boolean (1 = True) (0 = False)
useMap	Specifies what game map that server will use.	String Constant (FLANDERS, ALSACE, VERDUN,

		MARNE, ISLAND)
randomizeMap	Overrides the **useMap** setting and randomly picks a map each time the server is executed.	Boolean (1 = True) (0 = False)
regionSize	Restricts the aerodromes and landmarks to a region of a certain size around the center of the map. These sizes can be seen on the landmark bitmap files included with the server.	String Constant (FULL, LARGE, MEDUIM, SMALL)
landmarkDensity	Density of the landmarks and aerodromes on the map.	String Constant (FULL, DENSE, MODERATE, SPARSE)
minAerosPerSide	The minimum number of aerodromes per side.	Numeric
balanceLandmarks	Should the number of landmarks be balanced between the two sides, or the density and region size criteria be applied without regard for teams?	Boolean (1 = True) (0 = False)
gameType	Selects the game type for this server.	String Constant (TEAM_MELEE, MELEE, GET_THE_BARON)
gameMode	Selects the game mode for this server. Game mode affects the flight model, sun glare effects and weapons jamming.	String Constant (NORMAL, ADVANCED)
difficulty	Selects the game difficulty for this server. Game difficult will effect the number of bombs and rockets a player has, the accuracy of ground fire, and the amount of damage ground targets can withstand before being destroyed.	String Constant (ACE, VETERAN, NOVICE)
minKillsToScore	How many kills does a pilot have to have before they can be listed in the high score list for the following categories: bullets/kill, kills/sortie, points/sortie, and kills/death.	Numeric
startDate	Starting date for the server in the format: yyyy-mm-dd	Date
startTime	Starting time for the server in 24 hour format: hh:mm	Time
dayRate	Specifies the rate at which the day cycle will pass on the game server. This value represents a ratio. With a setting of 5, 5 game days pass in 1 “real” day. 1 game day cycle is approximately 16 game hours.	Numeric (Range: 1.0 – 1440.0)
nightRate	Specifies the rate at which the night cycle will pass on the game server. This value represents a ratio. With a setting of 100, 100 game nights will pass in 1 “real” day. 1 game night cycle is approximately 8 game hours.	Numeric (Range: 1.0 – 1440.0)
gameLengthInSeconds	Time length of this game. (A setting of zero means no time limit) When the time limit has expired and the game is over, the server will not restart automatically and must be restarted manually.	Numeric

HistoricAircraftDates	Specifies whether or not aircraft availability will be based historically. Starting with the starting date of the server and progressing through the war. Or if all planes will be available.	Boolean (1 = True) (0 = False)
hideTeamIndicators	Should the kneeboard display show or hide the center of action and/or center of team displays? (*Note: this also controls whether each baron appears on the kneeboard in GET_THE_BARON games.*)	Boolean (1 = True) (0 = False)
cloudsOn	Are clouds turned on? This will have a frame-rate impact on players without 3D acceleration.	Boolean (1 = True) (0 = False)
friendlyFire	Will fire from friendly planes cause damage to teammates.	Boolean (1 = True) (0 = False)
torqueOn	Are torque effects turned on?	Boolean (1 = True) (0 = False)
unlimitedAmmo	Do planes have unlimited bullets?	Boolean (1 = True) (0 = False)
rocketsAllowed	Are rockets and bombs allowed in the game?	Boolean (1 = True) (0 = False)
limitedFuel	Is fuel limited?	Boolean (1 = True) (0 = False)
maxPlaneTypes	Maximum number of different types allowed in the game at a time. For TEAM_MELEE games, this value will be divided in half for each team. *Example: with a value of 4, in a TEAM_MELEE game the maximum number of different planes available to each team will be 2.*	Numeric

Game Type Specific Settings

Get The Baron:

numBarons	How many people can be "it" in the game?	Numeric (Range: 1-8)
baronsGlow	Do baron planes glow at night? Barons will appear as a whitish gray color during the daytime and will glow at night.	Boolean (1 = True) (0 = False)

Team Melee/Target:

targetSwapTimeInSeconds	If the game type is TEAM_MELEE, and you want the server to select ground targets for the teams. Specify how long those targets are active before new targets are assigned. Setting this entry to 0 will assign no targets in the game. A maximum of 8 targets per team will be chosen, regardless of game size. For a total of 16 targets in the game.	Numeric
playersPerTarget	If the server is choosing targets, how many players per target will be assigned? *Example: With a setting of 4, if a team has 5 pilots they will be assigned 2 targets, and if they have 10 pilots, they will be assigned 3.)*	Numeric
targetRegionSize	How much of the map do you want to use for selected targets in a target game?	String Constant (FULL, LARGE, MEDIUM, SMALL)
endOnNoTargets	End the game when there are no more valid targets left. This setting will have no effect, unless **deadTargetsRemoved** is enabled.	Boolean (1 = True) (0 = False)
autoRestartServer	If **endOnNoTargets** is enabled, should the game reset when it is over?	Boolean (1 = True) (0 = False)
autoRepairTargets	Are landmarks repaired when they are selected as targets?	Boolean (1 = True) (0 = False)
deadTargetsRemoved	Are targets removed from the list of possible targets when they are destroyed.	Boolean (1 = True) (0 = False)
targetsChangeTeams	Do the targets witch from one team to the other, or do they stay with the team that gets them first? If this is set to false the targets may change but they will always be for the same team.	Boolean (1 = True) (0 = False)
germansDefendFirst	Which side starts with the targets assigned first?	Numeric (-1 = Random, 0 = Allied, 1 = German)
manuallyChooseTargets	Are targets specified manually? With this enabled, you can specify targets yourself rather than having the server choose them randomly. Note that if **landmarkDensity** is not set to FULL, then the targets you select may not appear on the map, since the landmark density is applied to all possible landmarks before targets are assigned.	Boolean (1 = True) (0 = False)
manualTargetsProgress	If **manuallyChooseTargets** is enabled, does the game "walk" through the manual target list? A setting of false will cause the targets to be chosen from the manual target list at random.	Boolean (1 = True) (0 = False)

manualTargetsAtATime	If **manualTargetsProgress** is enabled, how many targets do they get at once?	Numeric (Range: 1 – 8)

Plane Entries

The plane lists are divided by team. Each entry in the list has a plane entry followed by 3 settings in the following format:

PLANE_NAME = <*Availability*>, <*Starting Date*>, <*Ending Date*>

<*Availability*>	Specifies whether the plane is available on the server.	Boolean (1 = True) (0 = False)
<*Starting Date*>	Starting date of availability for this plane if **historicAircraftDates** is enabled.	Date (Format: yyyy-mm-dd)
<*Ending Date*>	Date at which the plane will no longer be available if **historicAircraftDates** is enabled.	Date (Format: yyyy-mm-dd)

Example:

MORANE_BULLET = 1,1915-8-1,1916-6-1

The [Targets] Section

If you want to manually set the targets for the game, you can list them in this section, by name. You may specify up to 8 targets per team. (*Note: See the text file "landmarks.txt" for a complete listing of the landmark and aerodrome names on all of the Red Baron 3D maps.*)

The [RegisterWith] Section

This section lists all of the meta-servers that your server will broadcast its availability to. Entries for this section are in the form of:

Protocol:IPAddress:Port

The *protocol* can be either UDP (for Red Baron meta-servers) or TCP (for WON.NET meta-servers). The *IPAddress* can be either the numeric address, or an alias. And finally, the *port* number is the port that the meta-server is running on.

Example: (These entries are just examples; meta-servers may or may not be running at these addresses)

UDP:dyn-meta.dynamix.com:47807
TCP:209.67.71.136:2667

The [Clients] Section

This section lists all of the "cookie" files that the server will use when performing security validation checking. The entries in this section are in the form:

Cookie=CookieXXXX.bin

Each cookie file name is based on the internal version of the client executable for which that cookie file was built. And new cookie files will be made available with any patches that are released.

Example:

Cookie=Cookie1002.bin