Operation Flashpoint:
Dedicated Server Administrator Guide

Installation

You need a Microsoft Windows 2000 or Windows XP computer to run an OFP dedicated server. We recommend using at least PIII 800Mhz machine, with 256Mb RAM and a 256kbps upload network bandwidth.

There are two possible ways to install the dedicated server. When you have installed the Operation Flashpoint game, you can use the main executable (*OperationFlashpoint.exe*) with the command line argument –server to start a dedicated server. While this is easy, it requires the Operation Flashpoint CD to be present in the drive during server operation. If you want to run a server without the CD, you have to use the standalone dedicated server executable. The latest version of this executable is always available from the official web site http://www.flashpoint1985.com, or as part of the most recent Upgrade also available from http://www.codemasters.com. To install the standalone server, you have to install the corresponding version of the game on any machine (you will need an Operation Flashpoint CD for this), run it at least once (to create a *Flashpoint.cfg* file) and add the *OFP_Server.exe* file to the same directory as OperationFlashpoint.exe. You can then copy the Operation Flashpoint directory on to the server.

This document describes dedicated server version 1.30.

Running the server

You can provide additional command line arguments when running a dedicated server:

-config=<config_file>	Select server configuration file (see below). Default: No configuration file; all parameters have default values.
-port=<port_number>	Select port at which session should operate. Default: 2234

In the event of a server crash, please help us to fix the problem by sending the *Flashpoint.rpt* and *context.bin* files that will be created in Operation Flashpoint directory on each crash. Please send those files to e-mail address support@bistudio.com; we will investigate them and if possible we will fix the bug that led to crash; or we will suggest you a workaround.

You may also consider running the OFP server as a service, and enabling automatic restart in case of crash. In this case, you may want to disable DrWatson crash monitoring utility on your computer, as it often prevents OFP server to shutdown properly (by displaying a message box that requires an operator to confirm application termination).

Server configuration

When running a dedicated server, you will usually want to create a Server.cfg file. In this file you can adjust many server parameters, and you can provide a mission list for automatic mission selection. An example can be found in Appendix B. The following entries are recognized in the configuration file:

password = <session_password>;	Password required to connect to server. Default: No password required.
passwordAdmin = <admin_password>;	Password required to administrate the server. Default: No password required.

motd[]=
{
"<1st MOTD line>",
"<2nd MOTD line>",
….
"<Last MOTD line>"
};
	Message of the day (MOTD). This message may consist of several lines. Each player is shown this text when connected to the server.
	
	Default: No MOTD.

motdInterval=<interval_in_sec>;	Interval in which subsequent lines of MOTD appear. Default: 5 seconds.
voteThreshold=<threshold>;	More than voteThreshold playes must agree when voting for some action. Default: 0.5 (more than half required).
reportingIP="<id_address">;	IP address of the master server to which this server is reporting its state. Use "" to disable reporting. Default: "master.gamespy.com".
voteMissionPlayers=<number>;	How many players must connect to the server for mission selection voting to start automatically. Default: 1

class Missions
{
class Mission01
{
template = <mission_name>;
cadetMode = <cadet_mode>;
param1 = <value>;
param2 = <value>;
};
class Mission02
{ … see above … };
….
class Mission<N>
{ … see above … };
 	
};
	Mission list
	
	First mission name
	Cadet mode (1) or Veteran (0)
	Values of mission-specific parameters. You will find more information about their meaning in the description.ext of the corresponding mission.
	
	When the end of the list is reached, the first mission is used again.

Note: The configuration file uses C++ like syntax. Each entry must be terminated with semicolon. You can also use C++ comments (starting with a double slash: //).

Performance tuning

There are also some parameters that can be used to fine-tune network performance. You can add following entries to Flashpoint.cfg (the main Flashpoint configuration file):

MaxMsgSend=<limit>;	Maximum number of messages that can be sent in one simulation cycle. Increasing this value can decrease lag on high upload bandwidth servers. Default: 26 Recommended for 256 kbps server: 64
MaxSizeGuaranteed=<limit>;	Maximum size of guaranteed packet in bytes (without headers). Small messages are packed to larger frames. Guaranteed messages are used for non-repetitive events like shooting. Default: 512
MaxSizeNonguaranteed =<limit>;	Maximum size of non-guaranteed packet in bytes (without headers). Non-guaranteed messages are used for repetitive updates like soldier or vehicle position. Increasing this value may improve bandwidth requirement, but it may increase lag. Default: 256
MinBandwidth =<bottom_limit>;	Bandwidth the server is guaranteed to have (in bps). This value helps server to estimate bandwidth available. Increasing it to too optimistic values can increase lag and CPU load, as too many messages will be sent but discarded. Default: 28800
MaxBandwidth=<top_limit>	Bandwidth the server is guaranteed to never have. This value helps server to estimate bandwidth available.

The greatest level of optimization can be achieved by setting the *MaxMsgSend* and *MinBandwidth* parameters. For a server with 256kbps we recommend the following values:

MaxMsgSend = 64;
MinBandwidth = 128000;

You can use the admin command *#monitor* to monitor server resource usage. The server never runs at more than 50 fps. When running slower, it always uses all available CPU processing power to maintain the smoothest possible gameplay. When running at less than 15 fps, you can consider the server overloaded – the mission currently played is probably too complex for given server. If you see the server is not using bandwidth that it could use, you can try increasing values *MaxMsgSend* and *MinBandwidth*.

Banning

To ban a user you have to know their unique online ID (called Player ID). This can be checked in the players overview screen ('P' key) during the game, or with the *#userlist* command. To ban a user you have to add their Player ID to file *ban.txt* residing in the main OFP directory. If there is no such file there, create a new one. The format of *ban.txt* is a plain ASCII text list of decimal Player IDs delimited with space, tabulator or end-of-line characters.

Appendix A: List of administrator commands

The following commands have special meaning when issued on global chat channel:

#login <password>	Login as server administrator
#logout	Logout, but stay connected as a normal user
#kick <player_name>	Kick given player
#kick <player_number>	Kick player with given number
#restart	Restart mission
#reassign	Go back to side selection screen
#mission <mission_name>	Select mission with known name
#missions	Select mission
#shutdown	Shutdown server
#userlist	Display list of all users
#monitor <interval_in_sec>	Start server monitoring. Server CPU load and bandwidth usage is displayed in the global chat channel. The default interval is 10 seconds. To stop monitoring type monitor 0.

Appendix B: Example of server.cfg file.

```
passwordAdmin = "xyzxyz"; // password to protect admin access
motd[]=
{
	"Welcome to OFP server.",
	"Hosted by Flashpoint Webworlds.",
}; // Welcome message
motdInterval=1;
voteThreshold=0.33; // when one third agrees, this is enough to confirm a vote
reportingIP=""; // private server - no reporting
voteMissionPlayers=3; // start voting when 3 players connect
```