Auto Installer Documentation

Brainless Technologies Ltd.

Table of Contents

Overview

The Ultimate Game Server Control Center (UGCC) is a web-based game server management panel. The auto installer feature of UGCC allows you to quickly deploy new game servers. This document details the required configuration and process.

A license is required to use the auto installer feature.

Contact Us

Need help or have questions?

- Join us on Discord: https://discord.com/invite/EFuewth
- Website/forums: https://www.brainless.us
- Other methods: https://www.brainless.us/contact.aspx

Quick start guide

This is a quick 'guide' on getting started with the automated game server installer using available templates.

1. Go to Game Types in settings page. Perform a Web Update Preview, review changes it wants to make. Then click the Web Update button.
2. Go to Installer Check in settings page. If any issues are identified go to the Server Definition page and fix settings identified. Return to Installer Check tool after making updates. Once it says everything is good, you may proceed.
3. Download an installer template (or create your own)
4. Import downloaded template using the Auto Installer Configs settings page. Review any notes.
5. Once imported go to the New Server page to deploy a new server
6. Review log tool and installer log for errors, then enjoy!

Sample installer templates

Some templates we've developed are available at http://redirect.brainless.us/UGCC/autoinstaller/templates/

Feel free to use them, customize them and learn from them.

High level process

This is a quick review of the process the auto installer uses to install new servers.

1. Find all ports in use using information in the database
2. Pick ports that may be available, and then verify with the operating system that they are available
3. Import the template into the database (variables are not translated) to get an ID for the new server

4. Create directory for new server
5. Update the server’s settings (see table for the settings in the template the installer overrides)
6. Update game and utility cache folder on target server
7. Run pre-execute commands (variables are translated)
8. Copy files to new server’s directory
9. Process config files (translating variables within the files)
10. Run post-execute commands (variables are translated)
11. Update any defined hashes in new server
12. The server is enabled and ready for use.

Configuration

Server Definition

1. Visit the Server Definitions management page, ensure values are set/accurate for all definitions for the following fields:
    a. Installer Settings
        i. Config file templates are stored under this folder.
    b. File Cache
    c. New Game Server location
    d. New Game Server location (alternate)
        i. The auto installer allows defining a second location to install new game servers through, this can technically be anything you want; i.e. a second drive, ‘fast’ storage, etc.
    e. Platform
    f. Location is accurate
        i. You may create additional locations with Locations management page
    g. Allow new game servers is checked if you want this server definition to be usable with new installs
    h. Ensure Offline is not checked
        i. Offline servers will not be considered by the auto installer as usable
    i. Ensure accurate usable IP(s) for the server are listed, these will be used by the auto installer.
        i. If this field is empty the installer will not consider this server definition as usable.
        ii. Do not use FQDN DNS addresses, you must use IP addresses. Using FQDN DNS address will cause port selection to fail.
    j. Max servers
        i. Allows you to limit the total servers the panel will deploy to this server using the auto installer.

Installer Check page

Go to the Installer Check page and verify the panel is happy with all server definitions.

Please note: Installer information is stored in <installer settings> on server definition 1. Don’t delete it.

Game Type management

In the Game Types management page, ensure all game servers you have already and/or want to deploy have values set for ports, Steam id's, etc. The more information you have about a game the better the auto installer will accomplish its job.

The installer will use this port information to make intelligent decisions on what ports to use if these default ports are already in use.

An entire section is dedicated to Game Type configuration later in this document.

Auto installer configuration settings

I know, this looks like a very long read, but some important information that doesn't really fit anywhere else is listed here. 😊

1. Visit the Auto Installer Configuration page in admin menu
    - k. Here you may setup and edit existing installer configurations, these are essentially templates for the installer.
2. To setup a new config
    - a. Click the new button on the bottom of the page, in the window that opens, enter a name for the install configuration and click the create button.
    - b. Choose the platform (Windows/Linux) this config is for
        - i. Configurations are typically platform specific and a game will more than likely require a separate config for each platform you want to support
    - c. Enter the template text (an example is provided further in the guide)
        - i. You can export an existing server's settings via the Import/Export tool in the admin menu. Once exported, copy and paste the contents of the downloaded file into the textbox.
        - ii. Variables in this template are not replaced by the auto installer. Reference the variable table later in this document
        - iii. A select number of settings will be overwritten by the installer, see the reference table further in the guide.
    - d. Set the game type for this server
        - i. To create a new game type, use the Game Type management page
        - ii. Existing game type information needs to be accurate and complete to assist the panel with choosing ports to use for new servers.
    - e. Configuration files define the files that the panel will process and copy to the new server location.
        - i. Enter one configuration file for each line
        - ii. Use relative paths from the server's root. Path separators don't matter, the panel will correct them depending on the target.
            1. For example, if you want to generate a server.cfg file that is normally in the root of the game server, you'd just enter 'server.cfg' on a single line
            2. For example, if you want to generate a server.cfg file that is normally in the <gameserver-root>/cfgs folder, you'd enter 'cfgs/server.cfg' on a single line.
        - iii. These files can use variables (it's recommended) and will be processed by the auto-installer. An example config file is included later in the guide.

iv. The configuration files are expected to be in the <installer settings>/<installer config ID> folder on the server definition with ID 1 (the server running the web panel). The relative path will be created as defined in installer config.

v. The panel will create empty files if they don’t exist in the above folder.

f. Pre-execute and Post-execute settings allow you to run custom commands prior to and after completion of the setup of new servers.

i. You should use variables

ii. Format is <executable>,<executable arguments>

iii. One command with its arguments per line

iv. If you need to use commas, quote the command and/or cmd arguments:

1. <command>,”arg1, arg2, arg3”
2. Parses as: <command>,arg1, arg2, arg3

v. If you need to use commas and quotes, escape the quotes with \:

1. <command>,”arg1, \”arg 2\”, arg3”
2. Parses as: <command>,arg1, “arg 2”, arg3

vi. If you don’t need commas, but do need quotes, escape them with \

1. <command>,\”arg 1\” \”arg 2\”

vii. Parses as: <command>,”arg 1” “arg 2”

viii. If you need to run a batch file (or other cmd built-ins (like copy, move, etc.) in windows, you’d use something similar to: c:\windows\system32\cmd,/c c:\path\to\my.bat

g. Select whether or not use SteamCMD. If the server can be installed via SteamCMD, set this to yes. Otherwise no.

i. If SteamCMD is set to yes, ensure the game type definition contains the SteamID for the game.

ii. If set to no, ensure you set the Cache Directory setting

h. For the Cache Directory setting, if you’re using SteamCMD, you may leave this field blank and the panel will create a cache directory using the Steam ID value found in the matching Game Type definition. For servers not using SteamCMD, this setting tells the panel where the files are located that it needs to copy.

i. Use a relative path (i.e. minecraft) and the panel will use the path in relation to <file cache> location, for example a setting of ‘minecraft’ would equate to ‘<file cache location>/minecraft’

ii. Again, path separators are not important. The panel will fix them if necessary.

iii. For servers that aren’t using SteamCMD, the panel expects this value to contain the files used to setup a new server. The installer will fail if the folder doesn’t exist.

i. Cache update command allows you to specify a command to run to update the cache if not using SteamCMD, or you need to run SteamCMD in a specialized way. Only used when ‘Use SteamCMD’ is no.

j. SteamCMD branch setting allows you to pass branch information to SteamCMD when updating the cache.

i. If you need to use +app_update 740 -beta, you’d add ‘-beta’ here

k. SteamCMD username setting allows you to define a username for steamcmd to login as when it updates the cache. If this is blank steamcmd with use anonymous.

i. Some game servers require a steam account.

l. SteamCMD password setting lets you define your password for above steam account.

i. Passwords are not exported in templates

m. Notes allows you enter notes about an installer config.
n. Click the save button

3. Other Misc Notes
    a. The panel will create blank config files in the <installer settings>/<id>/ folder if not found when saving your configuration settings. The configs will have their relative paths created.

Game Type configuration

Game type configuration is critical to successful use of the auto installer. The installer uses a lot of the information gleamed from the game type configuration about making decisions on port usage. Most settings should be self-explanatory, but we'll detail the others.

General settings

- Game type – This is a unique 'code' for this particular game type. It must be unique and 12 characters or less
- Query type – The protocol used when querying the server for map, player, etc. information
- RCON type – The protocol used when interacting with the game types' RCON implementation
- Default game port
    - The default port number and port type for the server.
- Default query port, Default query secondary port, Default RCON port
    - Similar in nature to the Default game port
    - Strict query port setting tells the installer if any deviation is allowed when creating new servers. We'll detail this later in this section.
    - A note about the Secondary Query port setting; most servers don't need this value. If you have a game server that includes its own integrated web panel (i.e. 7 Days to Die), or another port used for other purposes (Rust's mobile app support), you can specify them here, and then use the variable within the config files or command line options. Essentially if you don't need it for a game type, you are free to use it as you want.
- Other TCP ports, Other UDP ports
    - These list any other ports the game server may listen on, for normal purposes it's not relevant, but the installer will use this information to determine ports that are in use or could potentially be in use.
- NIC unbounded – If a game server is not able or does not provide the option of binding to a specific IP, set to yes. The installer will use this information to determine if the ports of this game type are tied to one IP address, or all IP addresses
- SteamCMD ID – The ID you would use with SteamCMD to install/update servers
    - The installer uses this for generating the cache
    - The %SteamID% variable value is also generated from this setting
- Steam APP ID, Workshop Support, Mod installation path
    - Are used by the panel's automated mod installer. Documentation is available for this at https://www.brainless.us/forum/viewtopic.php?f=21&t=2146 and this is not critical for the auto installer to function.

The strict port values setting for Query, Query 2, and RCON tell the panel if any deviation in port selection is acceptable. It's difficult to explain so I will provide some examples.

Counter-Strike: Global Offensive allows you define the game port value; we'll use the default of 27015. When it launches it also uses 27015 for RCON and query. If the installer were to pick different ports for query and/or RCON than the query and RCON port would not work within the panel. So in this case we'd set query and RCON as strict.

Minecraft has no expectation on port layout and thus strict would not be set on any of its ports. Minecraft provides a way to set all ports independently.

Another fictious example, a game server has a default game port of 2900 and the server always uses port +1 for query and port +2 for RCON and port +3 for some other function. The gametype would have strict set for query and RCON as that is not overridable via a configuration setting or command line option. With strict set on these ports, the installer will ensure the ports are incremented correctly.

If a game type (i.e. Minecraft) allows you define values for all ports, but you want to maintain a standard incrementation of ports (i.e. query is always +1, rcon is always +2, etc.) you are free to set the strict option.

In summary, if a game type allows you define all of the important ports you don't need to use strict port enforcement. If a game type automatically uses port that you can't define, you need to set strict port enforcement. If you want to enforce a strict port arrangement even if all ports are definable via a config or command line option, you can still set the ports to strict.

Graceful shutdown

If a game server is persistent between restarts and you want to save the game state before shutting down a server, you'll want to enable this feature.

- RCON say
    - The RCON command for public chat in the server. If %TEXT% is found the panel will replace %TEXT% with the text it wants to send.
    - If %TEXT% is not found, text to send will be appended to this setting's value
    - Used for timed restart/shutdown/update notices
- RCON save
    - If the panel should send an RCON command to initiate a world save, type the command here.
    - Can be blank
- RCON quit
    - Command the panel should send to stop the server. Most servers will do a world save before shutting down.

Note: If the panel doesn't have special functionality for finding the RCON password, you'll have to enter the password on the RCON tab for the server.

More in-depth information is available on our forums.
https://www.brainless.us/forum/viewtopic.php?f=21&t=637

Online update

The ‘Web Update’ button at the bottom of the Game Type administration page will pull a list of game type settings from our web server and add missing items and update existing entries. You must use the preview button first to preview the changes it will make. You should review the preview to make sure nothing changes that is important for your environment. Once the preview is viewed the Web Update button will be enabled.

If you customized any of the default game types, or created one with the same name as a new default, your settings will be overwritten. If you have a game type that you’ve modified, it’s best to save it with a different game type ID so the online updates don’t overwrite it.

We try to publish good data, but if you see anything incorrect or want to us to add a game type, simply contact us and we’ll do so after verification of the data.

Importing and exporting auto installer configurations

The export feature will export the associated auto installer configuration, game type configuration, and config files specified in the installer configuration into a single file. This allows the importation of a single file to get a complete setup for SteamCMD based games. For non-SteamCMD games, you simply need to populate the server cache directory specified by the install configuration.

Importing an auto installer configuration will add the auto installer configuration and recreate the specified configuration files. If the game type specified in the file doesn’t exist, the panel will import it as well. If the game type does exist and the checkbox ‘Import game type if it exists’ was selected the panel will import the game type with a new game type ID and also update the installer configuration file to reference the new game type ID. There’s no harm in having multiple game type definitions for the same game, but it may be a bit confusing.

Please note that games that have special graceful shutdown features, need to have a specific game type value. See the graceful shutdown section for further details.

Also note that SteamCMD passwords are not exported in templates.

Deploying a new server with the auto installer

To setup a new server using the auto installer, go to the New Server admin page.

Choose the Installer template, the user, location, server definition and number of player slots for the server.

As you select options, the panel will filter the usable options for you. For example, if you choose a template with a Linux platform, only Linux based server definitions will be available.

Click next and wait. You may also do a background install and the installation will continue in the background, the log will be shown in the server’s admin notes.

Once the installer completes, the log will be shown in the text box at the bottom of the page. The log will also be stored in the server’s admin notes field and added to the install.log file in server definition 1’s file cache location.

Logging

Logs for the auto installer are located in <installer settings>/installer.log.

A log for each install is also saved in the admin notes of each installed server.

Auto install order

The auto install completes these tasks in order:

- Find unused ports to use for new server
    - Searches the database, considers other UDP/TCP ports defined for each server's game type, and finally checks the server to determine if the port is in use by another process not managed by the panel.
- Import template (no variables are translated)
- Create new directory for server
    - Typically, <game server install location>/<owning user id>/<server id>
- Update server settings
    - See list of fields we override
- Update utility cache
    - Synchs files from server definition ID 1 cache\ugcc folder to target server
- Update cache
    - Synch files from server definition ID 1 or do a Steam update
- Run pre-commands
    - Variables translated
- Copy files from cache to new Game server directory
- Process/copy config files
    - Variables translated
- Run post-commands
    - Variables translated
- Enable server
- Finished

Supported variables

The most up to date list is available at https://www.brainless.us/forum/viewtopic.php?f=21&t=1030

These variables can be used in server settings, auto installer templates, auto installer pre/post commands, and auto installer config files.

Variable	**Description**
%Root%	Replaced with the associated server's root path setting
%ServerWD%	Replaced with the associated server's working directory setting
%DATE%	Replaced with current date
%TIME%	Replaced with current time
%IP%	Replaced with server's query IP address setting
%Port%	Replaced with server's game port setting

%Port+##% and/or %Port-##%	Replaced with the server’s game port + or - the number after the addition/subtraction sign. If %Port% would return 27015, %Port+5% would return 27020.
%RconPort%	Replaced with server's RCON port setting
%QryPort%	Replaced with server's Query Port setting
%QryPort2%	Replaced with server's Query Port 2 setting
%ServerID%	The server's ID
%OwnerID%	The ID of the server's owner
%SDefID%	The ID of the server's server definition (remote monitor)
%SDef%	The textual name of the server's server definition (remote monitor)
%Var1% - %Var12%	Custom Variable (Var1-12 are user definable per server and can be set to be editable by users. (See online documentation for more details)
%DependsID%	ID of the server the server depends on (Added in 1.862)
%Depends * %	Replaced with the variable of the server the server depends on (Added in 1.862 - See online documentation for details)
%SteamCMD%	Replaced with path and executable of steamcmd tool; useful for update tool. Panel and monitor will download steamcmd the first time the variable is encountered. (Added in 1.88)
%SteamID%	Replaced with the game type's defined steam id value. Added in 2.30. Useful for update command in server templates.
%RandomString%	Generates a random string. Useful for generating unique, random passwords in auto installer configs or if you want to set a unique password each time the server launches for RCON, etc.
%MaxPlayers%	Replaced with the Max Players setting of server.
%JavaHome%	Windows: Will search the registry and find the **path** of the latest version of Java installed. Linux: Blindly returns /usr/bin/java
%Utility%	Replaced with the server’s utility cache folder path (cache folder\ugcc)

Template settings the installer over-rides

The installer will replace these values in the template with values it determines during the install. Otherwise all other template values are used as specified in the installer’s template.

Name	Template Setting	Notes

Gametype	type	Game type setting from installer config is used and not game type in template.
Disabled		Set so the monitor nor users are able to start it before being finalized. The installer will enable the server when complete.
Query IP	qryip	
Game Port	gameport	
Query Port	qryport	
Query Port 2	qryport2	
RCON Port	rconport	
Server Root	serverroot	This will be replaced with the path the installer picked. Typically, <new game server path>\<User ID>\<Server ID>
Server Definition ID	sdefid	Target server definition
User	user	Owner of the server
Max Players Allowed	maxplayersallow	This is the player slot count specified when initiating the install.
Payment Due Date	duedate	Today’s date
Timed Restart Minute	restartmin	Randomized value (0, 10, 20, 30, 40, 50) so all servers don’t restart at the same time.
Command line	args, hiddenargs	If the installer finds the text ‘astrongpw’ in any command line, it replaces it with a random string. We use astrongpw for our RCON passwords in templates frequently.

Server file cache/updating workflow

The workflow differs a bit between whether SteamCMD is used or not.

Installs with SteamCMD

SteamCMD based caches should update fairly quickly and have pretty low overhead.

1. Create cache folder on target server
2. Run SteamCMD saving files in cache folder on target server
3. Copy cache folder files to new server location on target server
4. Process configuration files and queue job for target to process

Installs without SteamCMD

Non SteamCMD installs will have a bit of overhead for the initial synch and depending on the number and size of files could take a while. Future installs should involve less overhead and be quicker if the cache already exists on the target.

1. If Cache update command is defined, run the command to update cache.
2. On server definition 1 (sdef 1 is the computer the web panel is hosted from) check hashes on all files, queue a job for each file
3. Target server will compare its cache’s file hashes against what was submitted in the queue

4. If hashes differ, target will submit a job to the queue requesting the file
5. Sdef 1 will read the file contents, compress it and place data in the job queue
6. Target server will then download the file, decompress the data and save it to the appropriate file in the cache directory
7. Once the cache is up to date, a job will be queued to copy cache folder contents to new server location
8. Sdef 1 will process configuration files and queue job for target to process

The re-install tool feature

A reinstall button will be visible on the server management page if a server was deployed with the auto installer. The reinstall feature allows users to reinstall their game server. Users may reinstall their game server if

- The server is not in a disabled state
- The logged in user is the owner of the server
- The reinstall feature is not disabled
- It has been more than 24 hours since the last reinstall was performed

None of these restrictions apply to admins. The Admin Server Management page has its own Reinstall Server button.

For a reinstall the panel will create a new server, using the existing server as a template. It'll reuse the same ports and other settings from the old server and disable the old server. Once the install is complete the new server's ID will be changed to the old server's ID and the new server will get a new random ID. The only setting the panel doesn't carry over is the root path and command line argument settings.

Do note the paths may no longer match the default install location. For example the panel uses <userid>/<serverid> for new servers, a reinstalled server may have a path that doesn't match it's ID and that is to be expected.

If you want to prevent users from using the reinstall tool, enter the below text into the SQL tool. This will create an entry in the database disabling reinstalls for users and will require the panel to be restarted.

```
INSERT INTO config (config_name,config_value) VALUES ('disablereinstall',1)
```

To revert and allow reinstalls run the following in SQL tool.

```
DELETE FROM config WHERE config_name='disablereinstall'
```

The utility cache

The utility cache is a special folder that is synched each time the auto installer is used and was added in UGCC 2.47. It can also be synched to all server definitions via Settings->General configuration->Other tab.

If you have files or executables you need to ensure are available on each server definition you can use the utility cache folder for this purpose and target it with the %Utility% variable in pre/post commands and anywhere else it may be useful.

The job queue

The auto installer update introduced a new job queue. It utilizes the database as a message passing system to allow servers to send messages to each other utilizing the database.

If you run your database server on a slow server and/or with limited bandwidth, it could have a negative impact on the performance of non SteamCMD installs.

You can monitor the job queue via the Report Admin Tool. The tool has a few different reports to see the status of the job queue.

Status numbers used by the job queue are:

- -1 (Standby)
- 0 (Ready, not processed by target yet)
- 1 (target is actively processing the job)
- 2 (An error had occurred)
- 3 (OK)

The installer will try to repeat a failed job up to 3 times.

Automating the automated installer

Licensed users may use the API to deploy a new server. The basic API call is:

Panel.example.com/api.ashx?c=install&u=admin&p=password1&i=5&o1=5&o2=3&o3=16

Where u = A valid username of an admin account of the panel

Where p = password of above admin account

Where i = ID of installer configuration to use

Where o1 = ID of target server definition to place new server

Where o2 = ID of user to set as owner of new server

Where o3 = Number of player slots

Where o4 = Number of Game Server Installation Directory (optional, defaults to 0 for primary location, 1 to use alternate location. Will default to 0 if 1 is passed and is not configured.)

You may use the listusers (and/or newuser to create a new user), listsdefs, listinstallers API commands to get IDs of the various items setup with your panel. More information on the API is available at our website: https://www.brainless.us/forum/viewtopic.php?f=21&t=244

You may also use the uget tool to interact with the API either interactively or programmatically. More information is available at https://www.brainless.us/forum/viewtopic.php?f=21&t=1493

Troubleshooting

There are numerous places you should check incase the installer runs into issues.

- The installer's log file. Available in the admin settings menu under Installer Log.
- The panel's general log. The Log tool from the Admin Settings Menu.
- Job queue status. Errors will be reported from the job handler here. Use the report tool and select the 'Job Queue Data (Not Ok)' option.

Example config files

These are sample configuration files for a Minecraft Server. Note the use of variables, the installer will replace these variables with the respective setting for the server.

Auto Install Config Line	Contents (Variables will be replaced)

server.properties	`#Minecraft server properties`
	`#Fri Jul 31 14:12:38 EDT 2020`
	`spawn-protection=16`
	`max-tick-time=60000`
	`query.port=`**`%QryPort%`**
	`generator-settings=`
	`sync-chunk-writes=true`
	`force-gamemode=false`
	`allow-nether=true`
	`enforce-whitelist=false`
	`gamemode=survival`
	`broadcast-console-to-ops=true`
	`enable-query=true`
	`player-idle-timeout=0`
	`difficulty=easy`
	`spawn-monsters=true`
	`broadcast-rcon-to-ops=true`
	`op-permission-level=4`
	`pvp=true`
	`entity-broadcast-range-percentage=100`
	`snooper-enabled=true`
	`level-type=default`
	`enable-status=true`
	`hardcore=false`
	`enable-command-block=false`
	`max-players=`**`%MaxPlayers%`**
	`network-compression-threshold=256`
	`resource-pack-sha1=`
	`max-world-size=29999984`
	`function-permission-level=2`
	`rcon.port=`**`%RconPort%`**
	`server-port=`**`%Port%`**
	`debug=false`

	server-ip=**%IP%**
	spawn-npcs=true
	allow-flight=false
	level-name=world
	view-distance=10
	resource-pack=
	spawn-animals=true
	white-list=false
	rcon.password=**%RandomString%**
	generate-structures=true
	max-build-height=256
	online-mode=true
	level-seed=
	use-native-transport=true
	prevent-proxy-connections=false
	enable-jmx-monitoring=false
	motd=Powered by Brainless Tech Game Panel
	enable-rcon=true

Example template

An example template is included below, it is not complete and was edited to consume less space. It probably wouldn't work if you tried to import it.

Note the extended use of variables. The template needs to be as generic as possible so that I'll work regardless of where it's installed. Note some variables (like %TIME%) are specific to a function and won't necessarily be listed in the variables table. A list of variables was included earlier in this document.

Templates can be exported from existing servers or downloaded from our site and customized, re-exported and used for installer templates. My advice is to setup a server in the panel manually. Then edit it until you get to the point where all you have to edit are the IP address, ports, and root path settings on the query and executable tabs to setup a new server. Once you are at this point it's a good candidate to export and use with the auto installer.

We have made some templates available (see 'Sample Templates' section). Even if you don't plan to use any of the installer templates we've produced, they are good examples for how they should be setup.

[friendlyname]
Minecraft Server (Linux) - 25565
[/friendlyname]
[startpath]
%Root%
[/startpath]
[cmd]
%JavaHome%
[/cmd]
[cmdhash]

```
[/cmdhash]
[args]
-Xmx1024M -Xms1024M -jar server.jar nogui
[/args]
[hiddenargs]
[/hiddenargs]
[editargs]
0
[/editargs]
[type]
mc
[/type]
[allowupdate]
0
[/allowupdate]
[updcmd]
[/updcmd]
[updargs]
[/updargs]
[fedit]
0
[/fedit]
[fpath]
%Root%
[/fpath]
[qryip]
192.168.20.26
[/qryip]
[qryport]
25565
[/qryport]
[admin_notes]
[/admin_notes]
[autorestart]
1
[/autorestart]
[nostart]
1
[/nostart]
[priority]
4
[/priority]
[cpuafin]
0
[/cpuafin]
[AmountDue]
[/AmountDue]
[img]
[/img]
```

```
[timedrestart]
1
[/timedrestart]
[restarthour]
3
[/restarthour]
[restartmin]
0
[/restartmin]
[ftpserver]
[/ftpserver]
[ftpport]
[/ftpport]
[ftppassv]
0
[/ftppassv]
[ftpuser]
[/ftpuser]
[ftppath]
/
[/ftppath]
[FtpAbsPath]
%ServerWD%
[/FtpAbsPath]
[QryPort2]
0
[/QryPort2]
[restartday]
1
[/restartday]
[advrestart]
1
[/advrestart]
[RestartAlert]
0
[/RestartAlert]
[setschedule]
True
[/setschedule]
[cpumon]
0
[/cpumon]
[cputhreshold]
0
[/cputhreshold]
[cputime]
0
[/cputime]
[cpuaction]
```

```
0
[/cpuaction]
[pidfile]
[/pidfile]
[disabled]
0
[/disabled]
[gameport]
25565
[/gameport]
[dispip]
[/dispip]
[prvqryip]
[/prvqryip]
[timetostart]
1
[/timetostart]
[hash2]
[/hash2]
[hash2file]
%ServerWD%/server.jar
[/hash2file]
[secondaryexe]
[/secondaryexe]
[secondaryarg]
[/secondaryarg]
[rconport]
25575
[/rconport]
[steaminf]
[/steaminf]
[autoupdateusermin]
1
[/autoupdateusermin]
[rammon]
0
[/rammon]
[ramthreshold]
0
[/ramthreshold]
[ramtime]
0
[/ramtime]
[ramaction]
0
[/ramaction]
[maxplayersallow]
0
[/maxplayersallow]
```

```
[killchild]
0
[/killchild]
[deptime]
1
[/deptime]
[restartwhenempty]
0
[/restartwhenempty]
[limitcrashloop]
1
[/limitcrashloop]
[updonrestart]
0
[/updonrestart]
[restarthourly]
0
[/restarthourly]
[serverroot]
/ugcc/servers/mc
[/serverroot]
[alwaysautoupdate]
0
[/alwaysautoupdate]
[usebroker]
0
[/usebroker]
[updatehashes]
0
[/updatehashes]
[altautoupdate]
0
[/altautoupdate]
[console]
%Root%/logs/latest.log
[/console]
[restartinmin]
0
[/restartinmin]
[showres]
1
[/showres]
[envvars]
[/envvars]
[gracefuldelay]
10
[/gracefuldelay]
[gracefultext]
Server shutting down in %TIME%.
```

```
[/gracefultext]
[gracefultextnow]
Server shutting down now!
[/gracefultextnow]
[adminnotes]
[/adminnotes]
[updateextra]
[/updateextra]
[killparentpid]
1
[/killparentpid]
[rcongs]
0
[/rcongs]
[autorestartupdate]
0
[/autorestartupdate]
```