

# Crysis WebAdmin

A Server Web Administration Interface

By Fabian 'Fabifott' Schlieper

[fabi.me](http://fabi.me)

<b>1. INTRODUCTION</b>	<b>2</b>
<b>Features</b>	<b>2</b>
<b>2. SETUP</b>	<b>3</b>
<b>Crysis Server</b>	<b>3</b>
<b>Web Interface</b>	<b>3</b>
Windows	3
Linux	3
WebAdmin Settings	4
WebAdmin Console	4
<b>3. CREATING TEMPLATES</b>	<b>5</b>
<b>Special Expressions</b>	<b>5</b>
<b>Environment Variables</b>	<b>5</b>
<b>Public Classes Declarations</b>	<b>6</b>
public class TemplateBindings	6
public class TemplateBindingsRequest	6
public class TemplateBindingsServer	6
public sealed class WebUserInfo	7
public enum WebUserPrivileges	7
public class CrysisServerInfo	7
public class CrysisServerStatus	8
public class CrysisMapInfo	8
public class CrysisLevelRotationListEntry	8
public class CrysisPlayerInfo	9
public class CrysisServerConnectionInfo	9
public class CrysisServerBanInfo	9
public class CrysisChatEntry	9

# 1. Introduction

Crysis WebAdmin is a Web Administration Interface allowing you to administer your Crysis or Crysis Wars Server with a web browser.

It's a small standalone application that can either run on the game server machine or (with restricted functionalities) remotely on another server.

## Features

- multi-admin high secure web interface (IP range restriction, HTTPS, Sandboxing)
- individual admin privileges
- for Crysis and Crysis Wars
- server anti crash and scheduled auto restart<sup>1</sup>
- does not require any modifications to server files (apart from `autoexec.cfg`)
- fast template engine for customizable HTML pages
- automatic updates
- runs on Windows and Linux

With the web interface you can ...

- ... kick and ban players (bans can be given a note/reason)
- ... set game/server settings
- ... manage level rotation<sup>1</sup>
- ... upload new maps/map packs (ZIP & RAR<sup>2</sup>) from you local machine or an URL (HTTP & FTP)<sup>1</sup>
- ... use the chat and view the console<sup>1</sup>
- ... restart the server<sup>1</sup>
- ... manage web administrators and much more.

---

<sup>1</sup> This feature is only available if the web interface is running on the game server machine

<sup>2</sup> Uploading RAR archives requires WinRar to be installed (<http://www.rarsoft.com/download.htm>)

## 2. Setup

### Crysis Server

The web admin requires access to the external RCon HTTP Interface. To enable the interface add the following two lines to your `autoexec.cfg`:

```
http_password <password>
http_startserver port:8080
```

Replace `<password>` with a password that you later define in the WebAdmin Settings. Instead of port 8080 you can choose any port you like, except the Crysis game ports and ports already in use by you server.

For Linux: Commands like `g_nextlevel` or `map <map>` in your `autoexec.cfg` may cause trouble when the RCon HTTP Interface is enabled. The WebAdmin will call `g_nextlevel` automatically at the first connect, so just leave them out.

### Web Interface

#### Windows

Make sure you have at least .NET Framework 2.0 installed. Run `CrysisWebAdmin.exe` and proceed to [WebAdmin Settings](#).

#### Linux

To run Crysis Server Web Admin Interface under Linux the Mono Framework has to be installed. The following packages are required:

```
mono-jit
mono-gmcs
libmono-i18n2.0-cil
```

If you want to run the WebAdmin Interface in GUI mode you additionally need the Winforms package:

```
libmono-winforms2.0-cil
```

On Debian or Ubuntu you can get them with `apt-get`:

```
$ apt-get install mono-jit mono-gmcs libmono-i18n2.0-cil
```

On Suse you can use Yast. Visit <http://www.mono-project.com/> for more information.

Once you have installed Mono, make the files `start-gui` and `start-console` executable:

```
$ chmod 755 start-gui
$ chmod 755 start-console
```

That's it. Now you can either run the interface in GUI or console mode.

To start it with a nice GUI:

```
$ ./start-gui
```

Or to run in the shell:

```
$ ./start-console
```

See [WebAdmin Console](#) for information about the console mode.

## WebAdmin Settings

### Crysis Server Settings

In the Application GUI click on the tab "Crysis Server Settings".

Enter the game server IP address or hostname (if the web interface is running on the same machine it's localhost) and the RCon details you set in the `autoexec.cfg`.

If your game server is running on Linux, you should enable *Always keep RCon Connection alive*, since it doesn't accept RCon connection after the first one was closed.

Set the right server type and check the path settings.

The path of the server start file can be either set to the server's executable (e.g

`CrysisWarsDedicatedServer.exe` or `CrysisDedicatedServer.exe`) or a batch file (for Linux: a shell script).

If you specify the executable path you can additionally define start arguments.

Note that there is no server log file before the game server run the first time. So if cannot find a

`Server.log` (on linux `fp_log.txt`) just run server's executable and close it after it finished loading.

For Linux: the start file should be a shell script including a `cd` to the Crysis root path. For example:

```
#!/bin/sh
cd /path/to/CrysisWars
./Bin32/crysiswars-lnxserver -root "/path/to/CrysisWars" +exec "server.cfg"
```

If you want allow users to edit `server.cfg` or `autoexec.cfg` you can add these files to the text editor settings.

### Web Interface Settings

Now click the tab "Web Interface Settings". Define the desired web interface port (default HTTP port is 80). Only enable SSL if you have a valid SSL certificate installed! (Otherwise you'll get a blank page in your browser.)

Last step is to add web admin accounts. Click "Add admin..." and enter the admin details.

To test your server settings click "Test Crysis Server connection" after having started the game server. If the test succeeds, click "Start WebAdmin Interface" and change to the "Log" tab to see what is going on.

Open your web browser and browse to <http://localhost/> or `http://localhost:[port]/` if your HTTP listening port is not set to default 80.

### Common Settings

Here you can define the path to the WinRAR extractor command line executable (usually `unrar.exe`). If provided it is possible to upload maps and map packs in RAR archives. Otherwise, only ZIP is supported.

The option *Disable template cache* is useful for template creation since the templates will be compiled each request. Do not enable this when running on a live server, as it might cause security flaws.

If you want to test your template without running a game server enable *Template test mode*.

Clicking on the button View registered *WebAdminActions...* opens a dialog showing all registered admin actions and their required privileges. This is useful for template creation as well.

## WebAdmin Console

When running in GUI mode, all settings are easily accessible. In console mode (run WebAdmin with the `-console` parameter) it's not that comfortable:

With the first run of the WebAdmin the file `WebAdminSettings.xml` is created in the folder of the executable. This file stores all settings of the interface, including admin accounts.

You can either edit the file with a text editor or you can run the Web Interface in GUI mode (on Windows or Linux), adjust the settings and use the created `WebAdminSettings.xml`.

### Console commands

<code>start</code>	Start the Web Interface
<code>stop</code>	Stop the Web Interface
<code>adduser</code>	Add a user. Usage: <code>adduser username password [privileges]</code>
<code>quit, exit</code>	Exit the web interface

### 3. Creating Templates

First create a folder with a name of your choice in the `templates` directory. The name of the folder will appear in the template list. The easiest way to start is to copy all files from the default template to the new folder and building your templates on the existing one. (Warning: Do not edit the default template if you have automatic update enabled! The updater will overwrite all changed files the next time you start the interface.)

Don't forget to disable the template cache, as this offers a fast way to test the output.

The template language is based on C# using ASP-like code tags. Experiences in C-like programming language are recommended. A code tag looks like `<% [the C# code] %>`.

Code sections "`{ ...code... }`" can go across code tags, e.g.

```
<% if(trueOrFalse) { %>
... HTML Code ...
<% } %>
```

### Special Expressions

To include a template into another use the following code line:

```
<%@ Include File="file.inc.html" %>
```

To print a value in the HTML-code use this tag:

```
<%=someValue%>
```

`someValue` can be a variable of any type and will be converted into a string.

### Environment Variables

There are four environment variables accessible from the template code:

Name	Type	Description
Interface	<a href="#">TemplateBindings</a>	Provides common output functions
Request	<a href="#">TemplateBindingsRequest</a>	Information about the HTTP request
Admin	<a href="#">WebUserInfo</a>	Represents the logged in web administrator
Server	<a href="#">TemplateBindingsServer</a>	Provides functions to get data from the game server

All these variables are pointers to objects, so use the following syntax to access their members:

```
<%=Server.GetName() %> (This tag prints the server name)
```

Since this document does not explain the functions of the public class members it is best you take a look at the default template files before creating your own. Most of the member names are self-explanatory.

## Public Classes Declarations

Here you find the public declaration extracts of all classes accessible by the template code.

```
public class TemplateBindings
{
    public bool TextEditorEnabled { get; }
    public string[] TextEditorFiles { get; }

    public UserCommandLogEntry[] GetCommandLog();
    public UserCommandLogEntry[] GetCommandLog(WebUserInfo user);
    public string GetSupportedArchiveExtensions();
    public string GetTextFileContent(string file);
    public string HtmlEscape(object data);
    public string HtmlEscape(object data, bool parseColorTags);
}
```

---

```
public class TemplateBindingsRequest
{
    public string GetTextEditorFile();
    public WebUserInfo GetUser();
    public string GetVar(string name);
}
```

---

```
public class TemplateBindingsServer
{
    public bool AntiCrashEnabled { get; }
    public bool AutoRestartEnabled { get; }
    public bool IsCrysisWarsServer { get; }
    public bool IsResponding { get; }
    public bool IsRunningLocally { get; }

    public CrysisMapInfo[] GetAllMaps();
    public int GetAutoRestartInterval();
    public DateTime GetAutoRestartTime();
    public bool GetAutoTeamBalance();
    public string[] GetAvailableGameTypes();
    public CrysisMapInfo[] GetAvailableMaps();
    public CrysisServerBanInfo[] GetBans();
    public CrysisChatEntry[] GetChat();
    public CrysisLevelRotationListEntry[] GetConnections();
    public string[] GetConsoleLog();
    public byte GetCPUUsage();
    public string GetCurrentMapName();
    public string GetGameRules();
    public CrysisServerInfo GetInfo();
    public CrysisLevelRotationListEntry[] GetLevelRotation();
    public string[] GetMapDirectories();
    public CrysisMapInfo[] GetMapsToAdd();
    public string GetMaskedPassword();
    public int GetMaxPlayers();
    public int GetMemoryUsage();
    public int GetMinTeamLimit();
    public string GetName();
    public CrysisPlayerInfo[] GetPlayerList();
    public bool GetRandomizeLevelRotation();
    public int GetReviveTime();
    public CrysisServerStatus GetStatus();
    public CrysisPlayerInfo[] GetTeamPlayers(byte team);
}
```

```
    public byte[] GetTeams();
    public int GetTimelimit();
    public bool GetTKPunish();
    public int GetTKPunishLimit();
}
```

---

```
public sealed class WebUserInfo
{
    public DateTime LastAccessTime;

    public string Name { get; set; }
    public byte[] PasswordHash { get; set; }
    public WebAdminPrivileges Privileges { get; set; }

    public bool CanBan();
    public bool CanChangeGameSettings();
    public bool CanChangeGameType();
    public bool CanChangeMap();
    public bool CanChangeServerSettings();
    public bool CanDeleteMaps();
    public bool CanKick();
    public bool CanManageMapRotation();
    public bool CanManageUsers();
    public bool CanRestartMap();
    public bool CanRestartServer();
    public bool CanUploadMap();
    public bool CanUseConsole();
    public bool CanUseTextEditor();
    public WebAdminPrivileges[] GetInheritablePrivileges();
    public WebUserInfo[] GetJuniorUsers();
    public byte GetUploadPercentage();
    public TimeSpan GetUploadTimeLeft();
    public bool HasPrivilege(WebAdminPrivileges priv);
    public override string ToString();
}
```

---

```
[Flags]
public enum WebUserPrivileges
{
    None = 0,
    KickPlayers = 1,
    BanPlayers = 2,
    RestartMap = 4,
    ChangeGameType = 8,
    ChangeMap = 16,
    ManageMapRotation = 32,
    UploadMap = 64,
    DeleteMaps = 128,
    ChangeGameSettings = 256,
    ChangeServerSettings = 512,
    RestartServer = 1024,
    UseConsole = 2048,
    ManageUsers = 4096,
    UseTextEditor = 8192,
    All = 16383,
}
```

---

```
public class CrysisServerInfo
{
    public bool AntiCheat;
```

```
public bool Dedicated;
public bool DirectX10;
public bool FriendlyFire;
public string GameMode;
public bool GamepadsOnly;
public string GameType;
public string GameVersion;
public bool HaveData;
public IPAddress IPAddress;
public Exception LastError;
public string Map;
public byte MaxPlayers;
public string ModName;
public string ModVersion;
public string Name;
public byte NumPlayers;
public bool Official;
public bool PasswordProtected;
public int Ping;
public CrysisPlayerInfo[] Players;
public ushort Port;
public byte[] Teams;
public int TimeLeft;
public int TimeLimit;
public bool Updating;
public bool VoiceComm;
public CrysisPlayerInfo[] GetTeamPlayers(byte team);
}



---



public class CrysisServerStatus
{
    public readonly CrysisServerConnectionInfo[] Connections;
    public readonly string GameRules;
    public readonly string HostName;
    public readonly string LevelPath;
    public readonly int MaxPlayers;
    public readonly int NumPlayers;
    public readonly TimeSpan TimeRemaining;
}



---



public class CrysisMapInfo
{
    public readonly string Description;
    public readonly string[] GameTypes;
    public readonly string HeaderText;
    public readonly string MapPath;
    public readonly string Name;

    public override string ToString();
}



---



public class CrysisLevelRotationListEntry
{
    public readonly string GameRules;
    public int Id;
    public readonly CrysisMapInfo MapInfo;
    public readonly string MapPath;

    public override string ToString();
}
```

```
public class CrysisPlayerInfo
{
    public readonly int Deaths;
    public readonly int Headshots;
    public readonly int Id;
    public readonly int Kills;
    public readonly string Name;
    public readonly string Rank;
    public readonly byte Team;
}
```

---

```
public class CrysisServerConnectionInfo
{
    public readonly int Id;
    public readonly string IP;
    public readonly string Name;
    public readonly int Ping;
    public readonly int Profile;
    public readonly int State;
}
```

---

```
public class CrysisServerBanInfo
{
    public readonly int ProfileId;
    public readonly TimeSpan TimeLeft;
}
```

---

```
public class CrysisChatEntry
{
    public readonly string Message;
    public readonly string Receiver;
    public readonly DateTime Time;
    public readonly string Username;

    public override string ToString();
}
```