Star Trek - Voyager Eliteforce

Downloads

- Lilium Voyager Version 1.39, Win32, x86
- Lilium Voyager Version 1.39, Win32, x64
- Lilium Voyager Version 1.39, Linux, x86
- Lilium Voyager Version 1.39, Linux, x64
- Lilium Voyager Version 1.39, MacOS, x64
- Elite Force - Patch 1.38 (Thilo) (inzwischen offline)
- Elite Force - Installer inkl. Version 1.2
- Elite Force - Linux Dedicated Server 0.28 (Original Version)
- EF Log Analyzer
- EF bei United Federation of Planets
- EF bei The Last Outpost

Spiel-Info

EF iss ein wunderhübsches FP-Ballerspiel, thematisch im Star Trek Voyager Universum angesiedelt. Entwickelt wurde es von Ravensoft, veröffentlicht von Activision im September 2000. Das Spiel wurde für Windows, Linux, MacOS und sogar die Playstation 2 veröffentlicht. Die regulären Multiplayer-Spiel-Modi sind:

- Capture the Flag (CTF)
- Tournament (1 on 1)
- Deathmatch (FFA)
- Team Deathmatch (TDM)

Mit dem Addon, bzw. Version 1.2 des Spiels kamen zum Basisspiel weitere Variationen des Multiplayers hinzu:

- Elimination: Wer gefraggt wird, ist bis zum Beginn einer neuen Runde raus, muss also zusehen (ähnlich Counter-Strike).
- Specialties: Jeder Spieler spielt eine bestimmte Klasse die unterschiedliche Vor- und Nachteile hat. Es gibt keine Items auf den Maps. Der Infiltrator z. B. hat stets Speedboots, dafür aber nur einen Phaser. Seine Sonderfähigkeit ist einmaliges Wegbeamen. Der Techniker hat einen Bogenschweißer und ein Kraftfeld zur Verfügung. Es gibt die folgenden sechs Klassen:
  - Infiltrator
  - Sniper
  - Heavy weapons specialist
  - Demolitionist
  - Medic
  - Technician
- ActionHero: Ein Spieler startet mit mehr Gesundheit, die stetig regeneriert wird (+10 pro sek.), maximal bis auf 300. Das Schild-Level ist auch maximal 300 (wird aber wie regulär durch Aufsammeln aufgebaut und wird mit der Zeit weniger) und er hat alle Waffen zur Verfügung (aber nicht unendlich Munition). Er kriegt 1 Punkt je Frag. Die anderen kriegen 1 Punkt für

normale Fraggs. 5 gibt es für das Fraggen des ActionHero und der Fragger wird selbst zum ActionHero (die Fähigkeit wird vom Gefraggten dann abgegeben).
- Disintegration: Alle starten mit dem Federation Rifle mit unendlich Munition und es gibt *nur* den alternativen Feuermodus.
- Assimilation: Ein Spieler startet als Borg-Queen, deren Aufgabe es ist, die anderen (Federation) zu assimilieren. Jeder dazugewonnene Borg kann seinerseits assimilieren. Die Borg haben gewonnen, wenn alle Gegner assimiliert sind. Federation gewinnt, wenn die Borgqueen gefraggt ist.

Andere Mods: Es gibt auch noch eine Reihe anderer Modi, bzw. veränderter Modi, die aber nicht so häufig gespielt werden, z. B.

- Pinball: Ziel des Spieles ist es, den Gegner aus der Map zu schießen, nicht ihn durch reguläres Fraggen zu töten.

Hier mal ein erster Eindruck vom Gameplay des Spiels (Multiplayer auf Internet-Server):

- https://www.youtube.com/watch?v=UdH14lRiaKc
- https://www.youtube.com/watch?v=905qj7DFWHo

Das Spiel basiert auf der Quake 3 Arena-Engine, das heißt sehr viele der hier gesagten Dinge treffen auch auf Quake 3 Arena zu. Ebenso geht das dann anders herum. Das Spiel selbst ist unterteilt in den Einzelspieler-Modus und das „Holomatch“, also den Multiplayer. Alles, was hier steht, gilt vorrangig für das Holomatch, sprich den Multiplayer-Part des Spiels. Die meisten der Befehle und Einstellungen können aber auf beides gleichermaßen angewendet werden, bzw. wirken sich auf beide aus (z.B. die Helligkeitseinstellungen).

Das Spiel wurde von Activision am 14. September 2000 veröffentlicht, welche jedoch nach einigen Streitigkeiten mit Paramount (vertreten durch Viacom) die Lizenz für Star Trek Spiele im Allgemeinen zurück gaben. Aus diesem Grund werden Eliteforce und auch die anderen Star Trek Spiele von Activision nicht weiter vertrieben. Daher gibt es das Spiel auch nicht bei GOG oder anderen Retailern zu kaufen. Es bleibt daher nur der Gebrauchthandel, z. B. über eBay oder Amazon Market Place.

Systemanforderungen

Mindestsystemanforderungen

- Windows® 95, 98, 98 SE, NT 4.0 SP5, oder Windows 2000 mit 100% kompatiblem Computer System
- 3-D Hardware Beschleuniger mit vollem OpenGL® Support
- Pentium 233 MHz MMX® Prozessor und 8 MB Grafikkarte oder Pentium II 266 MHz MMX® Prozessor mit 4 MB Grafikkarte oder AMD® 350 MHz K6®-2 Prozessor mit 4 MB Grafikkarte
- 64 MB RAM
- Quad-Speed CD-ROM-Laufwerk (mind. 600 kB/Sek Transferrate)
- DirectX 7.0a oder höher
- Modem oder Netzwerkkarte für den Multiplayer

Empfohlene Systemausstattung

- Windows® 95, 98, 98 SE, NT 4.0 SP5, oder Windows 2000 mit 100% kompatiblem Computer System
- 3-D Hardware Beschleuniger mit vollem OpenGL® Support
- Pentium II 300 MHz MMX® Prozessor (oder schneller) mit 16 MB Grafikkarte
- 64 MB RAM (oder mehr)
- Quad-Speed CD-ROM-Laufwerk (mind. 600 kB/Sek Transferrate)
- DirectX 7.1 oder höher
- Dreitastenmaus mit Mausrad

Macanforderungen

- G3 oder G4 Prozessor mit 266 MHz oder schneller
- 64 MB RAM
- PCI oder AGP 3D Beschleuniger Grafikkarte mit mindestens 6 MB VRAM und Apple OpenGL Version 1.1.2 oder höher
- 560 MB unkomprimierter Festplattenspeicher (für virtuellen Speicher wird noch mehr Festplattenplatz benötigt)
- Mindestens 128 MB virtueller Speicher
- MacOS 8.6 oder höher
- 4x CD-ROM
- Internet (TCP/IP) und LAN (TCP/IP) Multiplayer wird unterstützt
- Internet Multiplayer verlangt mindestens ein 28.8 kbit/s Modem

Versionen

Es gibt von EF verschiedene Versionen. Die niedrigste, ausgelieferte Version ist 0.28. Aktuell ist die neuste offizielle Version 1.2 und die letzte inoffizielle (via Lilium-Voyager) ist 1.39. Um die Version sicher festzustellen, kann man einfach kurzerhand ein Solo- oder Multimatch starten. Sobald der Server läuft, ESC drücken und auf `Server Data` klicken. Es wird daraufhin in der Liste der Variablen die Version mit angezeigt. (Es gibt auch im Hauptmenü eine Angabe, welche aber mit gewissen `*.pk3`-Dateien überschrieben werden kann, sodass man sich darauf nicht verlassen kann.)

Bekannte Versionen sind:

Version	Netzwerkversion*	Kommunikationsprotokoll**	Versionsstring	Bemerkung
0.28	22	-	`ST:V HM v0.28 win-x86 Aug 17 2000`	Erste von Raven veröffentlichte Version, unter Win 10 ist der Singleplayer nicht lauffähig.
1.1	23	-	`ST:V HM v1.10 win-x86 Nov 10 2000`	Zweite, von Raven veröffentlichte Version, Singleplayer funktioniert jetzt auch unter Win 10.
1.2	24	-	`ST:V HM v1.20 win-x86 Apr 17 2001`	Dritte und letzte offizielle Version. Kommt in zwei Varianten, einmal nur das Standard-Spiel auf Version 1.2 gepatcht, und einmal als Expansion-Pack-Version. Die Versionsangaben sind aber dieselben.

Version	Netzwerkversion*	Kommunikationsprotokoll**	Versionsstring	Bemerkung
1.35	24	-	`ioST:V HM v1.35 win_msvc-x86 Jul 17 2006`	Version von Thilo Schulz, basiert auf ioQuake 3
1.36	24	-	`ioST:V HM v1.36 win_msvc-x86 Aug 26 2006`	Version von Thilo Schulz
1.37	24	-	`ioST:V HM v1.37 win_msvc-x86 Nov 13 2006`	Letzte von Thilo Schulz als Release veröffentlichte Version.
1.38 RC 1 x86	24	26	`ioST:V HM 1.38_SVN2130M win_mingw-x86 Aug 3 2011`	Letzte von Thilo als Release Candidate veröffentlichte Version.
1.38 RC 1 x86 x64	24	26	`ioST:V HM 1.38_SVN2130M win_mingw64-x64 Aug 3 2011`	Letzte von Thilo als Release Candidate veröffentlichte Version.
1.39 x86	24	26	`Lilium Voyager HM 1.39_GIT_23b94816-2019-05-12 win_mingw-x86 May 12 2019`	Aktuelle Version von Lilium-Voyager, basiert auf ioQuake 3. Wird von einem gewissen Zack "zturtleman" Middleton gepflegt. Der Versionsabschnitt `_GIT_23b94816-2019-05-12 […] May 12 2019` kann abhängig vom git-Revisionsstand und Compile-Datum anders aussehen.
1.39 x86 x64	24	26	`Lilium Voyager HM 1.39_GIT_23b94816-2019-05-12 win_mingw64-x86_64 May 12 2019`	Aktuelle Version von Lilium-Voyager, basiert auf ioQuake 3. Wird von einem gewissen Zack "zturtleman" Middleton gepflegt. Der Versionsabschnitt `_GIT_23b94816-2019-05-12 […] May 12 2019` kann abhängig vom git-Revisionsstand und Compile-Datum anders aussehen.

* Das ist die Versionsnummer (`protocol`), die letztlich übereinstimmen muss bei Server und Client.
** Das ist letztlich die genauere Angabe welches Kommunikationsprotokoll (`com_protocol`) verwendet wird. So weit bisher festzustellen, scheint das aber keine größere Auswirkung zu haben, solange nur das `protocol` übereinstimmt. Allerdings scheint EF 1.2 ohne Expansion-Pack 1.39er Server im lokalen Netz automatisch zu finden, während 1.2 ohne EP und älter die Server gar nicht anzeigen.

Die Patches 1.1 und 1.2 verlangen im Prinzip nur eine Vorbedingung: EF muss klassisch installiert worden sein, da sonst notwendige Registry-Keys nicht vorhanden sind, sodass der Patch die Anwesenheit des Spiels verneint (und einen nur abbrechen lässt).

EF einrichten

Installation

Zuerst sollte man das Spiel zum Laufen zu bringen. Heutzutage ist das nicht mehr völlig selbstverständlich, wenn auch das Spiel selbst ziemlich zuverlässig läuft. Es müssen einige Voraussetzungen erfüllt sein: Die Quake 3-Engine und damit das gesamte Spiel, basiert auf OpenGL, das heißt das ihr zum Spielen auf jeden Fall OpenGL mit euren Grafiktreibern installiert haben solltet. Wer also nur die Standard-Treiber von Windows benutzt, könnte die Erfahrung machen, dass das Spiel sich ggf. weigert zu starten, weil OpenGL nicht gefunden wird. Abhilfe schaffen normalerweise die neuesten Treiber vom Grafikkartenhersteller, in denen das enthalten sein sollte. Zur Not kann man es aber auch mit den auf der CD enthaltenen OpenGL-Treibern versuchen (nur für Windows). Dazu müsst

man auf der Installations-CD von EF im Verzeichnis `\Setup\GlSetup` die Datei `glsetup.117.exe` starten. Die funktioniert aber nicht bei jedem, also lieber gleich nach den eigenen Graka-Treibern suchen. Die Hersteller helfen da meistens aus. CD-Keys sind bei dem Spiel eigentlich automatisch durch den Installer integriert, werden also nicht ab gefragt. Sollte es aber aus irgend welchen Gründen mal Schwierigkeiten damit geben, sollte man sich seinen Key ggf. vorher notieren oder sichern. Sonst heißt es neu installieren/kopieren. Dedicated Server fragen den Key aber nicht ab und machen auch keinen Abgleich, ob sie frei im Internet zu finden sind. Man kann also eigentlich irgend einen nehmen, solange er nur echt ist.

Das Spiel selbst benötigt wenige Dateien von der CD. Die Key-Datei `efq3.key`, die Archiv-Datei(en) `pak0.pk3` (und ggf. `pak1.pk3` und `pak2.pk3` für Version 1.2) im `baseEF`-Ordern, sowie die Dateien `stvoy.exe` und `stvoyHM.exe` im Stammverzeichnis werden von der CD benötigt. Auf manchen Systemen werden ebenfalls die DLLs `FFC10.dll`, `efuix86.dll`, `efgamex86.dll` und `binkw32.dll` für den Singleplayer im Stammverzeichnis benötigt. Für das Expansion Pack kommen noch die Dateien `pak3.pk3` und `expefq3.key` dazu. Ggf. auch noch Thilos Binaries oder Lilium Voyager. Die `*.pk3`-Dateien gehören ins Unterverzeichnis `baseEF` der EF-Installation, der Rest ins Stammverzeichnis. Mit Thilos Binaries oder Lilium-Voyager könnten diese Dateien auch im entsprechenden Verzeichnis des Homefolders liegen. Da sie bei allen Spielern auf einem Rechner gleichermaßen vorausgesetzt werden, bietet es sich trotzdem an, sie im Spielverzeichnis abzulegen. Die `*.key`- bzw. `*.dll`-Dateien und die Binaries (`*.exe`) gehören dagegen ins Stammverzeichnis von EF. Für Thilos Binaries bzw. Lilium-Voyager bitte die entsprechenden Abschnitte weiter unten beachten.

In Zeiten von Win64 und Drittanbieterpatches ist die klassische Installation nicht immer ganz so einfach. Deshalb hier ein kurzer Abriss, auf welchen Wegen man EF über den Installer installiert bekommt:

- Windows
  - 64 Bit
    1. `CD-Laufwerk:\Setup\Setup.exe` ausführen (weil der Starter unter Windows 64 Bit nicht mehr läuft)
    2. Das Expansionpack installieren (dieser Starter funktioniert ganz normal)
    3. Neue Binaries installieren (eines von beiden reicht):
       - Die Binary Dateien für Lilium Voyager herunterladen (siehe Downloads) und ins Spieleverzeichnis entpacken. zum Spielen reichen die Dateien `liliumvoyhm.x86_64.exe`, `liliumvoyhm_renderer_opengl1_x86_64.dll`, `liliumvoyhm_renderer_opengl2_x86_64.dll` und `SDL264.dll`. Falls man auch einen Dedicated Server aufstellen können möchte, benötigt man auch die `liliumvoyded.x86_64.exe`.
       - ~~nach~~ http://thilo.kickchat.com/efport-progress/bin/win64 ~~gehen und~~ die Dateien `iostvoyHM-1.38_rc1.x64.exe`, `SDL64.dll` und `renderer_opengl1_x64.dll` ins EF-Verzeichnis herunterladen. Die `*.dll`-Dateien könnte man prinzipiell auch ins Windows-Verzeichnis (`%windir%`) speichern, allerdings braucht man dazu Admin-Rechte und ob es irgendwelche Seiteneffekte gibt, ist mir nicht bekannt.
  - 32 Bit
    1. EF installieren
    2. Das Expansionpack installieren
    3. Neue Binaries installieren (eines von beiden reicht):
       - Die Binary Dateien für Lilium Voyager herunterladen (siehe Downloads) und

ins Spieleverzeichnis entpacken. zum Spielen reichen die Dateien `liliumvoyhm.x86.exe`, `liliumvoyhm_renderer_opengl1_x86.dll`, `liliumvoyhm_renderer_opengl2_x86.dll` und `SDL2.dll`. Falls man auch einen Dedicated Server aufstellen können möchte, benötigt man auch die `liliumvoyded.x86.exe`.
- ~~nach~~ http://thilo.kickchat.com/efport-progress/bin/win32 ~~gehen und~~ die Dateien `iostvoyHM-1.38_rc1.x86.exe` und `renderer_opengl1_x86.dll` ins EF-Verzeichnis herunterladen. Die Lib-SDL 1.2 (*nicht* die 2er!) für 32 Bit herunterladen und die im Archiv befindliche `SDL.dll` ins EF-Verzeichnis kopieren. Die `*.dll-Dateien` könnte man prinzipiell auch ins Windows-Verzeichnis (`%windir%`) speichern, allerdings braucht man dazu Admin-Rechte und ob es irgendwelche Seiteneffekte gibt, ist mir nicht bekannt.

  - Probleme mit Win 10: Seit kurzem gibt es wohl ein Problem bei der Installation unter Windows 10. Zwar startet der eigentliche Installer, aber er kommt bei 100% einfach nicht weiter. Dem kann man wie folgt abhelfen:
    1. Taskmanager öffnen
    2. Den Task des Installer Launchers mit dem Pfeil links ausklappen
    3. Rechtsklick auf einen der Unterpunkte, und dann `Zu Details wechseln` wählen.
    4. Auf den dann markierten Prozess nochmal rechtsklicken und `Warteschlange analysieren` auswählen
    5. Es findet sich stets ein Prozess, der den Installer-Prozess aufhält. Dieser ist dann zu beenden.
    6. Sollte sich beim ersten ausgewählten Installer-Prozess noch nichts tun, kann man bei den beiden anderen Prozessen dasselbe versuchen. Findet sich da nichts mehr, sollte der Installer bereits einfach weiter laufen. Quelle
  - Alternativ: Man kann EF auch manuell von den CDs runter kopieren. Benötigt werden die Dateien `pak0.pk3` bis `pak3.pk3` (Abhängig von Version und Verwendung des Expansion Packs). Diese gehören in einen Ordner `baseEF` kopiert. Eine Ebene darüber sollten die Dateien `FFC10.dll`, `efuix86.dll`, `efgamex86.dll` `binkw32.dll`, `stvoy.exe` und `stvoyHM.exe` liegen, mit denen der Singleplayer bzw. der Multiplayer gestartet wird, sowie die CD-Keys `efq3.key` bzw. `expefq3.key`. Die Dateien liegen auf den CDs jeweils im Ordner `Setup`.
- Linux
  - Prinzipiell sollte für beide der Linux-Installer verwendet werden können. Die Ausführung erfolgt via Dateiname. Dazu muss das File ggf. erst noch das Executable-Recht gesetzt bekommen.
  - Entweder man hat bei der Installation die CDs mit samt Installer zur Hand oder man kopiert die Dateien wie oben beschrieben manuell ins EF-Verzeichnis. Die so erzeugte Installation ist allerdings nicht für den Single-Player brauchbar. Dazu wäre Wine notwendig, um EF klassisch zu installieren: Wie für Windows beschrieben und dann auch via Wine spielen.
  - 64 Bit:
    1. Binaries besorgen (eines von beiden reicht):
       - Die Binary Dateien für Lilium Voyager herunterladen (siehe Downloads) und ins Spieleverzeichnis entpacken. Zum Spielen reichen die Dateien `liliumvoyhm.x86_64`, `liliumvoyhm_renderer_opengl1_x86_64.so`, und `liliumvoyhm_renderer_opengl2_x86_64.so`. Falls man auch einen Dedicated Server aufstellen können möchte, benötigt man auch die `liliumvoyded.x86_64`. Sie gehören ins Spielverzeichnis. Die `liliumvoyhm.x86_64` bzw. `liliumvoyded.x86_64` müssen ggf. noch

ausführbar gemacht werden.
- Von http://thilo.kickchat.com/efport-progress/bin/linux die Dateien `renderer_opengl1_x86_64.so` und `iostvoyHM-1.38_rc1.x86_64` ins EF-Verzeichnis laden. Letztere ausführbar machen.

2. `sudo apt-get install libsdl1.2-dev` oder etwas vergleichbares für die eigene Distribution eingeben, um die lib-SDL 1.2 zu installieren (es braucht *nicht* die 2er).

◦ 32 Bit
1. Binaries besorgen (eines von beiden reicht):
   - Die Binary Dateien für Lilium Voyager herunterladen (siehe Downloads) und ins Spieleverzeichnis entpacken. Zum Spielen reichen die Dateien `liliumvoyhm.x86`, `liliumvoyhm_renderer_opengl1_x86.so`, und `liliumvoyhm_renderer_opengl2_x86.so`. Falls man auch einen Dedicated Server aufstellen können möchte, benötigt man auch die `liliumvoyded.x86`. Sie gehören ins Spielverzeichnis. Die `liliumvoyhm.x86` bzw. `liliumvoyded.x86` müssen ggf. noch ausführbar gemacht werden.
   - ~~Von~~ http://thilo.kickchat.com/efport-progress/bin/linux die Dateien `renderer_opengl1_i386.so` und `iostvoyHM-1.38_rc1.i386` ins EF-Verzeichnis laden. Letztere ausführbar machen.
2. `sudo apt-get install libsdl1.2-dev:i386` oder etwas vergleichbares für die eigene Distribution eingeben, um die lib-SDL 1.2 zu installieren (es braucht *nicht* die 2er). Es kann ggf. notwendig sein, für die 32-Bit-Version noch die Libraries `libvorbisfile3:i386`, `libcurl3:i386` und `libmad0-dev:i386` zu installieren. Zumindest für die 32-Bit 1.37er Version sind sie notwendig.

◦ Ggf. noch die folgenden Libraries nachinstallieren, sofern das Spiel nicht starten möchte: `libmad`, `libogg`, `libvorbis`, `libvorbisfile`, `libopenal` und `libcurl`.

Das Expansionpack ist nicht zwingend nötig, aber hat einmal den Vorteil, dass die 1.2er Patches (für Windows) schon dabei sind. Zum anderen fügt es eine Reihe ganz guter Maps hinzu. Zum Spielen des Singleplayer braucht man dann allerdings die EP-CD. Der kursierende Singleplayer-Crack funktioniert nicht mehr! Der Multiplayer-Crack geht weiterhin.

Mit dem 1.38er Patch (basiert auf der ioquake3-Engine) ist ein Multiplayer-Crack aber nicht mehr nötig, denn da ist die CD-Abfrage-Routine schon raus. Der Patch ist nicht vom Hersteller von EF, sondern privaten Ursprungs, behebt aber einige Bugs bzw. Sicherheitslücken. Zusätzlich gibt's ein paar neue Spielmodi. Aber ich finde, schon alleine wegen der fehlenden CD-Abfage rentiert er sich.
 Zum Mitspielen ist der 1.38er Patch nicht unbedingt nötig, auch 1.2er Versionen können auf 1.38er Servern mitspielen. Anders rum aber auch. Gleiches gilt im Prinzip für Lilium Voyager.

Probleme mit CD-Keys

Zu kurzer CD-Key

Es gab offenbar einige CDs, auf denen ein fälschlicherweise zu kurzer CD-Key mit nur 17 Zeichen integriert war. In diesen Fällen hilft es, eine `0` anzuhängen, falls EF danach fragen sollte. Falls schon ein anderer eingegeben wurde, im Ordner `BaseEF` der EF-Installation die Datei `efq3.key` öffnen und dort als 18. Zeichen die `0` hinzufügen.

Kein CD-Key dabei, trotz Abfrage

Wenn man die EF-Version aus dem „Star Trek: Action Pack“ installiert, kommt diese Abfrage. Man kann hier den Dialog abbrechen, da der Installer den CD-Key danach mit installiert.

Hunk_Alloc failed

Dieser Fehler tritt auf, wenn Elite Force zu wenig Arbeitsspeicher zur Verfügung hat, bzw. umgekehrt, zu viele Maps, Models und sonstige Erweiterungen geladen werden. EF benutzt von Haus aus nicht allen verfügbaren RAM. Das Limit kann man selbst hoch schrauben: Entweder man gibt in die Konsole `/seta com_hunkmegs x` ein, mit x als die Menge Arbeitsspeicher, die EF verwenden soll. Oder man setzt `seta com_hunkmegs x` in die Konfig. Alternativ kann man natürlich auch seinen EF-Ordner aufräumen und nicht mehr benötigte Maps & Co. entfernen. Es reicht dazu aber auch aus, die seltener genutzten `*.pk3`-Dateien in einen anderen Ordner zu verschieben, auf den EF nicht zugreift (z. B. Unterordner `Backup` anlegen).

Grafikprobleme mit ATI-/AMD-Grafikkarten auf neuen Systemen

Die OpenGL-Implementierung verursacht bei AMD-Grafikkarten inzwischen öfter Probleme mit alten Spielen. Es gibt offenbar einen Workaround, der hier beschreiben wird.

Ort der Konfig-Dateien

Auch wenn es zu Beginn ggf. etwas verwirrend ist, gibt es nicht nur genau einen Ort, an dem die Einstellungen für das Spiel abgelegt werden. Bei einer Standardinstallation von CD, ohne Mods oder neuere Versionen als 1.2, liegen die Dateien unter `baseEF` im EF-Spiele-Ordner oder ggf. in den entsprechenden Ordnern der Mods im Spieleverzeichnis. Für den Mod Pinball ist das dann z. B. statt des Unterordners `baseEF` der Ordner `pinball`. Mit den Binaries von Thilo oder Lilium-Voyager sieht die Sache noch ein mal anders aus. Denn hier wird (wie es sich eigentlich für die Original Version von Raven gehört *hätte*) im Benutzer-Verzeichnis. Das hat den Vorteil, dass jeder Benutzer seine eigene Konfiguration und installierte Mods, Maps, Models, etc. haben kann und kein Benutzer Schreibrechte auf dem Spieleverzeichnis braucht.

Thilos Binaries

Nachdem die Quake 3 Engine Open Source wurde, haben sich auch sogleich einige daran gemacht, Quake 3 zu patchen und um Funktionen zu erweitern. Diese Gelegenheit hat auch ein gewisser Thilo Schulz genutzt und für EF Patches geschrieben, die darauf basieren. (Heute empfiehlt es sich, lieber gleich Lilium Voyager zu verwenden, einfach weil dort die Arbeit fortgesetzt wird, die Thilo inzwischen eingestellt hat.) Ein kurzer Abriss an praktischen Änderungen:

- EF auch für Linux
- Der Gamma-Bug ist behoben (zu dunkles Bild)
- Map-Downloads können jetzt vom Client auch via curl geladen werden (deutliche Beschleunigung)

- Einstellungen liegen jetzt im Benutzer-Ordner, nicht im Spieleordner (so kann jeder Spieler am PC seine eigenen Einstellungen setzen)
- Man kann mit ALT + TAB aus dem Spiel tabben.

Siehe auch pak92.pk3.

Hier findet man momentan den Quellcode und dort vorkompilierte Binaries für die Systeme.

Unter Linux liegen die Konfigs in `$HOME/.stvef/baseEF` (bzw. den entsprechenden Mod-Verzeichnissen). Unter Windows 7 liegen sie unter `%appdata%\STVEF\baseEF` bzw. den entsprechenden Mod-Ordnern. Abhängig von der Windows-Version (z. B. XP kann es auch an einem anderen Ort im Benutzerverzeichnis liegen), aber der Name STVEF sollte zu finden sein. Hier liegen dann, neben den Mod-Daten, ggf. auch Maps und Models. Allerdings existiert das Verzeichnis erst nach einem ersten Start von EF 1.34 oder höher.

Lilium Voyager

Was ist das?

Da Thilo irgendwann aufgehört hat, seine EF-Patches zu pflegen, gibt es von ihm keine neueren Versionen mehr, als die 1.38 RC 1. Und auch diese ist inzwischen nicht mehr auf seinen Seiten zu finden. Seine Arbeit wird von Zack Middleton in Form des Projekts „Lilium Voyager“ fortgeführt.

Wo zu erhalten

Es gibt die Quell-Dateien hier. Vorkompiliert findet man sie auch oben in den Downloads.

Wo sind meine Einstellungen hin?

Die Einstellungen werden für die drei großen Systeme, Windows, Linux und MacOS jeweils hier gespeichert:

- Linux: `~/.local/share/lilium-voyager/baseEF`,
- Windows: `%appdata%\Lilium Voyager\baseEF`
- MacOS: `~/Library/Application Support/Lilium Voyager/baseEF`

bzw. in den entsprechenden Mod-Ordnern.

Selbst bauen

Wie oben geschrieben, werden nur die Quelldateien veröffentlicht. Kompilieren muss man sich die ausführbaren Dateien („Binaries“) selbst. Wie das für Quake 3 und abgeleitete Spiele (32 und 64 Bit, verschiedene Plattformen) geht wird auf den Unterseiten dieser Seite gut beschrieben. Es ist aber zu beachten, dass die Beschreibung für ioQuake3 gedacht ist, also die git-Links nicht für Lilium-Voyager sind, sondern für ioQuake3. Für die Quelldateien von Lilium-Voyager bitte diese Quellen verwenden. Für Cross-Compiling von Linux nach Windows ist MingW notwendig. Ein beherztes `sudo apt-get`

`install mingw-w64` ist ggf. vorher notwendig (für Debian & abgeleitete Distributionen...).

Grafik-Einstellungen

Zu den einzelnen Grafikeinstellungen in `Grafikdaten` gibt es nicht viel zu sagen. Die Wirkungsweisen sollten bekannt sein. Allerdings ist der im nächsten Abschnitt beschriebene Helligkeits-Regler in `Grafikdaten II` zu finden, falls noch etwas nachjustiert werden muss.

Helligkeit einstellen

EF hat mit den original Binaries unter NT-Systemen (also alles ab Win2000/XP aufwärts) so seine Probleme mit dem Gamma-Wert. Soll heißen, das Spiel ist meist einfach viel zu dunkel, egal was man bei der Installation oder im Menü mit dem Schieberegler eingestellt hat. Das kann man wie folgt ändern:

1. Die Datei `efconfig.cfg` im Verzeichnis `BaseEF` der EF-Installation öffnen,
2. die Zeile `r_ignorehwgamma "X"` suchen und X auf `0` setzen,
3. die Zeile `r_overBrightBits "X"` suchen und auch hier X auf `0` setzen.

Dann ist es zwar eigentlich zu hell, aber lieber zu hell und ich sehe etwas, als zu dunkel und ich sehe nichts. Das oben genannte gilt natürlich auch für die Datei `hmconfig.cfg`, die den Effekt entsprechend für den Multiplayer bereitstellt. Für Thilos Binaries oder Lilium-Voyager ist diese Einstellung nicht nötig, da hier der Schieberegler genau das tut, was er soll.

Andere Auflösungen nutzen

EF basiert auf der Quake 3 Arena Engine. Damit lässt sich hier auch der Q3A Grafik Hack anwenden und EF auch in HD bzw. mit **Wide-Screen-Auflösungen** zu spielen. Die beiden relevanten Konfig-Files sind `efconfig.cfg` (für den Single-Player) und `hmconfig.cfg` (für den Multiplayer), siehe auch Abschnitt Ort der Konfig-Dateien. Für Mods (z. B. Pinball) muss ggf. die jeweils im Mod-Order abgelegte Konfig ebenfalls angepasst werden.

Username und andere Einstellungen

Username einstellen

Eine Unsitte, die auch im Netz gerne mal auftritt ist, dass Spieler vor Multiplayer-Spielen vergessen, ihren Namen einzustellen. Das kann unter Umständen ganz lustig sein, wenn 3 Spieler mit den Namen `Redshirt` auf einmal in der Liste stehen. Also am besten sofort ändern, dann vergisst man das auch nicht. Dazu geht man in EF in `Konfigurieren` → `Spieler` und dort kann man dann den Namen und das Model der Spielfigur, das man benutzen möchte, wählen.

Farbige Nicknames

Man kann auch Farben im Nick verwenden, indem man ein ^ mit einer Ziffer voranstellt. Alle nachfolgenden Zeichen werden dann in der entsprechenden Farbe dargestellt. Die Farbzuweisung ist wie folgt:

Code	Farbe
^1	Rot
^2	Grün
^3	Gelb
^4	Blau
^5	Cyan
^6	Pink
^7	Weiß
^8	Schwarz

Überlange Nicks

Von Haus aus ist die Länge des Nicks in EF durch das Eingabefeld limitiert. Wenn man aber die `hmconfig.cfg` manuell editiert (Zeile mit `seta name "..."`), sind auch längere Nicks möglich.

Handicap

Die Option `Handicap` ist insofern nur für ganz besonders gute Spieler interessant, alsdass sie die Schilde-, Gesundheits- und Waffenschadenswerte, auf den eingestellten Prozent Wert reduziert (inkl. oberer Grenzen). Ein Handicap von 50 bedeutet also, dass man jedes mal nach dem Spawning nur 50 Health-Punkte hat, nicht 100 wie normalerweise und auch höchstens auf 100 aufladen kann (mit langsamem Abfallen zurück auf 50). Das wirkt sich auf die selbe Weise auf die Schilde aus. Es wird als Basis einfach immer der Normalwert x das Handicap : 100 angenommen, das heißt normale Hyposprays laden ebenfalls nur auf den Handicap-Wert auf, die Booster-Hyposprays, gelbe Sanitasche, sowie Schilde laden zwar wie sonst auch auf einen Wert von maximal 200 auf, dieser verringert sich aber immer wieder langsam bis auf den Handicap-Wert (es sei den man hat Feindkontakt, dann meistens schneller... 🙂). Wie gesagt eher was für ganz gute Spieler, die das Spielerlebnis nicht für alle anderen versauen wollen.

Spieloptionen

Hier stehen noch ein paar Grafikeinstellungen des Spiels, die Teils sehr wichtig sind:

- `Lichtspiegelungen`: Ganz klassische Grafikeinstellung
- `Trefferspuren`: Ist ganz lustig an Wänden, aber eigentlich ein reines Gimmick
- `Dynamische Lichter`: Ganz klassische Grafikeinstellung
- `Zielinfo`: wird der Name des Gegners gezeigt, wenn ich ihn anvisiere, oder nicht?
- `Bilder Synchronisieren`: Wenn `ein`, haben alle Gegner das selbe Modell, das man selber benutzt. Das ist sehr gut, wenn jemand im Netz ein Modell benutzt, das man nicht installiert

hat. Dann sieht man ihn ohne diese Option einfach nicht.
- `Team-Overlays`: Blendet an der gewünschten Stelle Health und Schilde der eigenen Team-Kameraden ein
- `Autom. Download`: Sollen fehlende Maps, Models oder Mods automatisch vom Server runter geladen werden?
- `Einfache Objekte`: Sind Gegenstände wie Waffen oder Munition 3D-Objekte oder nur 2D-Bitmaps (schaut in 2D echt schlecht aus)?
- `Text`: Kann `Deutsch`, `Englisch` und `Französisch` sein.
- `Dialoge`: Können `Englisch` oder `Deutsch` sein.

Steuerung

Auch die „Steuerung“ bedarf evtl. einer Anpassung. Ein wichtiges Merkmal eines guten Ballerspiel ist es, dass man die Steuerung auch an die eigenen Bedürfnisse anpassen kann, so auch in EF.

- `Waffen`: passt von Haus aus ganz gut
- `Angriff/Umsehen` ist sehr wichtig, weil man dort so Kleinigkeiten einstellen kann wie `Alt. Angriff` auf die rechte Maustaste legen. Aber das ist Geschmacksache.
- `Kommentare`: Damit haut man Chat-Messages raus. Sollte man schon haben, weil die Konsole unhandlich ist.
- `Maus/Joystick` ist recht wichtig für Leute mit Gamepad, weil man da die Empfindlichkeit einstellen kann. Wird aber selten verwendet.
- `Weitere Optionen`: Nicht sonderlich wichtig, außer man will schleichen (`Immer Rennen` auf `aus`).

HUD

Das HUD zeigt im Prinzip alle relevanten Informationen an. Die Standard-Ansicht (also ohne Konfigurationsänderungen wie Net-Graph & Co.) sieht so aus:



Die einzelnen Elemente sind folgende:

1. Tragbares Item, siehe auch Ultritium Mine ff.,
2. Platz für aktive Items, ggf. werden hier mehrere zugleich angezeigt,
3. ein aktives Item, siehe auch Passiv genutzte Gegenstände ff.,
4. Restzeit, bis das Item direkt rechts daneben seine Wirkung verliert,
5. Score, hier können Kills, Deaths oder auch Flaggen-Captures stehen,
6. Rest-Munition, gibt nicht zwingend die Restanzahl Schüsse wieder, siehe auch Waffen,
7. Health-Bar, kann von 0 (tot) bis 300 (wird langsam bis auf 100 abgebaut) stehen,
8. Schilde (können von 0 (keine) bis 200 (wird langsam bis auf 100 abgebaut) stehen,
9. Name der aktuellen Waffe, wird nur eingeblendet, wenn man gerade auf diese Waffe gewechselt hat,
10. Waffenauswahl, ausgekreuzte (8) sind Munitionslos, das hell-violette (2) ist die aktive Waffe, der Rest sind verfügbare Waffen und
11. Chat-Bereich, hier werden die Sprachmeldungen aber auch Server-Meldungen ausgegeben.

Waffen und Items

Es gibt in Elite Force ein Set an Standardwaffen und eine handvoll versteckter Waffen (die zum Teil in Mods benutzbar sind, oder mit Cheats). Hier gibt es eine Übersicht:

Waffen

Praktisch alle Waffen haben zwei verschiedene Feuermodi. Der normale Modus liegt auf der linken Maustaste, der alternative auf ALT. Letzteres wird üblicherweise aber von jedem Spieler recht bald auf die rechte Maustaste gelegt. Teils unterscheiden sich die Feuermodi ganz erheblich, teils unterscheiden sie sich letztlich nur in Feuerkraft und -rate (und Munitionsverbrauch). Waffen lassen sich von Haus aus mit dem Scrollrad und ihren Zifferntasten durch schalten. Waffen mit Flächenschaden (Splash-Damage) können einem auch selbst schaden. Die Respawn-Zeit der Waffen

selbst wird serverseitig festgelegt. Die Respawnzeit der Munition ist 40 Sekunden. Der Schaden lässt sich durch einen Schadensfaktor serverseitig verändern (z. B. 2 bedeutet doppelt so viel Schaden) und durch Beeinflussung des Doubledamage-Faktors (kann z. B. auch vierfach sein, statt doppelt).

Phaser



Der Phaser ist quasi die Minimalausstattung. Er hat zwar vergleichsweise wenig Feuerkraft, dafür lädt sich dessen Munition von allein wieder auf. Es gibt daher auch kein Minutions-Item dafür. Der Alternative Modus verursacht mehr Schaden, zieht die Energie aber auch schneller leer. Auf große Entfernungen ist er weitestgehend nutzlos, weil er viel zu ungenau trifft. Mittlere und kurze Entfernungen sind kein Problem.

- Startmunition: 50
- Maximalmunition: 50
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 1
- Munitions-Item enthält: 0 (es gibt keines, Waffel lädt von selbst auf)
- Taste: 1

Kompressionsgewehr





Das Kompressionsgewehr verursacht schon erheblich mehr Schaden, vor allem im alternativen (Sniper-)Modus. Letzterer wird im Disintegration-Modus zum einzigen Modus. Gerade im Sniper-Modus kann man damit noch recht gut auf große Entfernung treffen. In der Kampagne hat das Gewehr zusätzlich die Eigenschaft, im Sniper-Modus keinen Lärm zu verursachen.

- Startmunition: 32
- Maximalmunition: 128
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 8
- Munitions-Item enthält: 32
- Taste: 2

U-Mod





Der U-Mod ist gewissermaßen die Schwesterwaffe des Kompressionsgewehrs, mit dem Unterschied, dass er als einziger die Schilde ignoriert (auch tragbare Kraftfelder!), also direkt auf die Gesundheit durch schlägt, statt zuerst die Schilder zu reduzieren. Alle anderen Waffen reduzieren zuerst die Schilde auf null und ziehen danach erst Gesundheit ab. Wie beim Kompressionsgewehr ist der Alternative Modus deutlich stärker, kostet aber auch mehr Energie.

- Startmunition: 15
- Maximalmunition: 60
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 3
- Munitions-Item enthält: 15
- Taste: 3

Scavenger Gewehr



Das Scavenger-Gewehr lässt sich am ehesten mit einer Maschinenpistole (streut stark, vergleichsweise langsame Ausbreitung der Kugeln) vergleichen, die im alternativen Modus Plasmabursts ballistisch schießt (Granatwerfer, prallt nicht ab). Der Granatwerfer erreicht höchstens mittlere Reichweiten. Der Primärmodus streut insbesondere auf größere Entfernungen stark, der sekundäre hat Splash-Damage.



- Startmunition: 30
- Maximalmunition: 100
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 5
- Munitions-Item enthält: 30
- Taste: 4

Stasiswaffe



Entspricht am ehesten einer Schrot-Flinte im primären Feuermodus. Auf kurze Distanz ist sie sehr effektiv, auf weite nicht, da die abgefeuerten Wirbel relativ langsam unterwegs sind. Im alternativen Modus feuert sie fünf sich verteilende Strahlen, ähnlich dem Kompressionsgewehr oder dem U-Mod. Der mittlere trifft dabei ähnlich gut, wie der U-Mod.



- Startmunition: 15
- Maximalmunition: 50
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 1
- Munitions-Item enthält: 15
- Taste: 5

Granatwerfer



Der Granatwerfer ist genau richtig benannt. Der primäre Modus schießt ballistische Granaten, die auch von Wänden und anderen Hindernissen abprallen. Der alternative Modus ist ähnlich, nur dass die Granate hier nicht abprallt, sondern als Mine am Hindernis haften bleibt. Sie löst aber nach



einem kurzen Moment aus. Die Flugbahn ist flacher aber gerader, als im primären Modus. Die Reichweite ist relativ begrenzt. Im Specialties-Modus bleiben die Minen dauerhaft aktiv und sind mit einem Laser versehen. Wird dieser von einem Spieler durchquert, löst die Mine aus. Damit lässt sich ein Raum effektiv mit Fallen versehen. Die Schusslinie ist dann gerade. Minen können sich auch gegenseitig auslösen, wenn sie zu nah zueinander platziert werden. Der Splash-Damage beider Modi kann einen auch selbst treffen. Der Granatwerfer wird nicht automatisch genutzt, wenn man ihn aufsammelt.

- Startmunition: 10
- Maximalmunition: 30
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 1
- Munitions-Item enthält: 10
- Taste: 6

Tetryon Puls Disruptor





Der Tetryon Puls Disruptor lässt sich im primären Feuermodus am ehesten mit einem Maschinengewehr vergleichen. Hohe Feuerkraft, Strahlfeuer, aber eine gewisse Streuung. Auf große Distanzen trifft man damit sehr schlecht. Interessant ist der alternative Modus vor allem dann, wenn das Ziel um die Ecke liegt, da die abgefeuerten Kugeln einige male abprallen, ähnlich den Granaten des Granatwerfers, nur ohne Auswirkungen der Gravitation. Die Feuerrate ist dabei aber erheblich größer.

- Startmunition: 40
- Maximalmunition: 120
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 2
- Munitions-Item enthält: 40
- Taste: 7

Photonenstoß





Diese Waffe entspricht den klassischen Raketenwerfern, wie man sie in Quake 3 Arena oder Unreal Tournament 2004 findet. Der primäre Modus feuert einen Torpedo, mit relativ viel Schaden und relativ hoher Fluggeschwindigkeit. Der Splash-Damage ist ausgesprochen hoch. Der alternative Modus ist ähnlich. Der Torpedo hat mehr Feuerkraft, verfolgt sein Ziel mit einer begrenzen Wendigkeit, aber breitet sich langsamer aus. In beiden Fällen ist ein Ausweichen noch relativ gut möglich. Die Waffe wird nicht automatisch genutzt, wenn man sie aufsammelt.

- Startmunition: 6
- Maximalmunition: 20
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 2
- Munitions-Item enthält: 6
- Taste: 8

Bogenschweißer



Der Bogenschweißer feuert gewissermaßen einen Blitz, mit relativ hohem Schaden. Der alternative Modus schießt eine Art Kugelblitz, der hohen Schaden anrichtet und einen größeren Wirkungsquerschnitt hat. Auf mittlere bis große Entfernungen ist die Waffe nicht sehr genau.



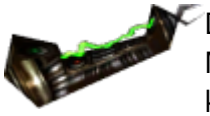
- Startmunition: 40
- Maximalmunition: 120
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 5
- Munitions-Item enthält: 40
- Taste: 9

Assimilator



Der Assimilator ist nur im Assimilation-Modus für Borg-Spieler verfügbar. Beide Modi assimilieren den Gegner. Er ist eine reine Nahkampfwaffe, ohne Munition oder Kosten. Aktiviert wird er mit der Taste 1.

Borg Waffe



Die Borgwaffe wird in der Kampagne von den Borg verwendet, im Multiplayer nur von Mods. Der Assimilation-Mod nutzt ihn für das Borg-Team. Der primäre Modus feuert kleine Plasmakugeln, ähnlich denen des Scavenger Gewehrs, der sekundäre eine Art Blitz.

- Startmunition: 100
- Maximalmunition: 100
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 1
- Munitions-Item enthält: 0 (Waffe lädt sich von allein wieder auf, es gibt keine Munitions-Items)
- Taste: 2

Andere aktiv genutzte Gegenstände

Tricorder

Der Tricorder ist nur mit dem Expansionpack verfügbar. Mit dem primären Modus kann man Objekte scannen. Der alternative zeigt ein Radar.

- Taste: `t`.

Hypospray

Das Hypospray als benutzbarer Gegenstand ist im Specialties-Mod ein Heilgegenstand, den Spieler mit der Rolle des Medic an anderen Spielern anwenden können. Der Unterschied beim Heilen für alternativen und primären Modus liegt in der Höhe des Gesundheits-Boost für den Empfänger.

- Startmunition: 50 für primäre Heilung und 30 für sekundäre Heilung.
- Munitionskosten Primärmodus: 1
- Munitionskosten Sekundärmodus: 1
- Munitions-Item enthält: 0 (es gibt dafür kein Munitionsitem)
- Taste: h

Ultritium Mine

Das Pendant zum Redeemer von Unreal Tournament ist eine kleine Sprengladung, die man zunächst legen muss, und die man dann fernzünden kann. Sie kann aber, ähnlich dem Redeemer, vorher vom Gegner zerstört werden, ohne dass sie explodiert. Der Splash-Damage ist ausgesprochen hoch und in etwas größerem Radius sorgt die Mine noch für eine Verschiebung der Spieler auf der Karte.

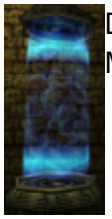
- Respawnzeit: 1 Minute

Kraftfeld

Kraftfelder können nur von Spielern des eigenen Teams (sofern vorhanden) durchquert werden. Sie bleiben eine Minute lang bestehen. Im Pinball Mod wird die Wirkung dahingehend verändert, dass keiner das Kraftfeld berühren darf, weil er sonst sofort stirbt. Das Kraftfeld kann durch Waffen frühzeitig zerstört. Ein einzelner Photonentorpedo mit Doubledamage zerstört es zum Beispiel sofort.

- Dauer: 25 Sekunden
- Respawnzeit: 1 Minute

Notfalltransporter

Der Notfalltransporter beamt den Spieler an einen zufälligen Spawn-Point der Karte. Im CTF-Modus lässt er dabei die Flagge fallen.

- Respawnzeit: 1 Minute

Med-Kit


Lädt die Gesundheit des Spielers wieder auf 100 auf. Ist er schon bei 100, erhält er 25 dazu.

- Respawnzeit: 1 Minute

Passiv genutzte Gegenstände

Es gibt diverse Items, die einem Boni oder Sonderfähigkeiten verleihen. Manche davon sind zeitlich begrenzt in ihrer Wirkung. Verliert man letztere, setzt derjenige, der das Item anschließend wieder aufsammelt, beim selben Countdown fort, wo der Vorbesitzer aufgehört hat.

Quantenverstärker


Ein klassischer Double-Damage Booster. Der übliche Standard-Wert ist Verdoppelung der Schadenswirkung für den Träger, was sich aber mit cvars anpassen lässt.

- Dauer: 30 Sekunden
- Respawnzeit: 2 Minuten

Seeker Drone


Feuert im Sekundentakt auf in der Nähe befindliche Gegner.

- Dauer: 30 Sekunden
- Respawnzeit: 2 Minuten

Jetpack


Das Jetpack lässt den Spieler frei Fliegen. Springen wirkt dabei als nach oben fliegen, Ducken als nach unten fliegen. Ansonsten wird mit den normalen Richtungs-Tasten und der Maus gesteuert.

- Dauer: 30 Sekunden
- Respawnzeit: 2 Minuten

Speedboots


Machen den Träger doppelt so schnell, nicht nur beim Laufen sondern auch bei der Feuerrate. Hält 30 Sekunden an.

- Dauer: 30 Sekunden
- Respawnzeit: 2 Minuten

Tarnfeld


Macht den Spieler nahezu unsichtbar.

- Dauer: 20 Sekunden
- Respawnzeit: 2 Minuten

Metaphasische Schilde


Macht den Spieler unverwundbar.

- Dauer: 20 Sekunden
- Respawnzeit: 2 Minuten

Nano-Regenerativer Schild


Lädt die Gesundheit des Spielers in Schritten von 5 pro Sekunden wieder auf.

- Dauer: 30 Sekunden
- Respawnzeit: 2 Minuten

Notfall Hypospray


Das Notfall Hypospray lädt bis zu 25 Gesundheit auf ein mal auf, ist aber bei 100 Gesamt-Gesundheit gedeckelt. Hat man schon 100 Gesundheit, wird es nicht mal aufgenommen.

- Respawnzeit: 30 Sekunden

Booster Hypospray


Das Booster Hypospray lädt 5 Gesundheit wieder auf, bis zu einem Wert von maximal 200 (für Action-Heroes 300). Hat man den Maximalwert erreicht, wird es nicht aufgenommen. Da (außer bei Action-Heroes) die Gesundheit dann aber von allein wieder pro Sekunde eins weniger wird, nimmt man letztlich praktisch doch gleich wieder eines auf.

- Respawnzeit: 30 Sekunden

Schild klein


Lädt 5 Schilde auf, bis maximal 200.

- Respawnzeit: 20 Sekunden

Schildrüstung dunkel


Lädt 50 Schilde auf, bis maximal 200.

- Respawnzeit: 20 Sekunden

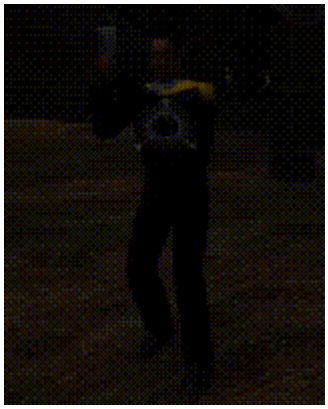
Schildrüstung hell


Lädt 100 Schilde auf, bis maximal 200.

- Respawnzeit: 20 Sekunden

Schild-Schaden vs. Gesundheitsschaden

Alle Waffen verursachen Schaden, das ist trivial. Aber nicht alle Waffen verursachen immer die gleiche Art Schaden. Abgesehen vom offensichtlichen Unterschied zwischen Flächenschaden bei Granaten und Co. und zielgerichteten Waffen gibt es z. B. den kleinen Detail-Unterschied beim U-Mod, der die Schilde einfach komplett ignoriert (auch tragbare Kraftfelder!). Allgemein gilt: Zuerst werden die Schilde reduziert. Dann erst wird die Gesundheit angegriffen. Schild-Schaden sieht aus der Sicht des Angreifers so aus:



Man sieht die Einschlagwolke des Gewehrs und den Bewegungseffekt auf das Spieler-Model, aber nicht mehr. Aus der Sicht des Angegriffenen sieht das so aus:



Es werden beim Gewehr-Treffer 18 Schild abgezogen, aber die Gesundheit wird nicht beeinflusst. Das gilt im Übrigen auch für Flächenschaden, wenn er denn nicht die ganzen Schilde aufbraucht:



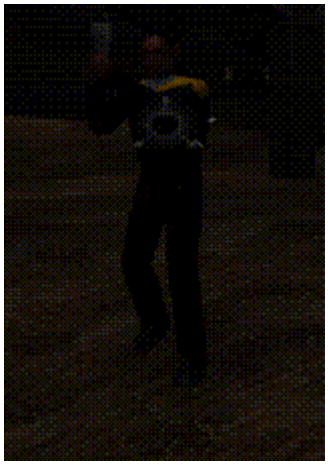
Der Schaden beträgt hier 78 an den Schilden. Die Waffe schlägt aber noch nicht auf die Gesundheit durch. Erst bei 0 Schilden wird die Gesundheit reduziert. Das sieht aus Sicht des Angreifers so aus (U-Mod ignoriert Schilde):



Man sieht deutlich die hellen Glitzer-Effekte zusätzlich zu den anderen Effekten. Aus der Sicht des Angegriffenen sieht das so aus:



Der U-Mod macht 23 Schaden an der Gesundheit, während die Schilde gar nicht beeinflusst werden. Man erkennt aber auch ganz ohne Werte-Anzeige am glitzernden Waffen-Model, dass man selbst gerade Schaden an der Gesundheit einsteckt. Mit Schilden sieht man davon nichts. Der Übergang kann besonders gut beim Phaser gesehen werden, da dieser vergleichsweise wenig/langsam Schaden anrichtet:



Bzw. die Werte:



Zuerst werden die restlichen Schilde aufgebraucht, danach die Gesundheit. Gerade für Waffen mit besonders viel Schaden (oder mit einem Quantenverstärker aktiv) kann es aber sein, dass man schon mit einem einzigen Treffer die Schilde beseitigt und auch gleich noch einen Gesundheitsschaden verursacht. Die Animation entspricht dann direkt dem Gesundheitsschaden.

Empfehlungen/Achievements

In EF gibt es diverse Errungenschaften, die man erreichen kann. Diese werden teils schon während dem Spiel, teils erst am Ende des Matches angezeigt bzw. bekanntgegeben. Hier ist eine vollständige Liste dieser Achievements:

Während dem Spiel

Symbol	Titel	Beschreibung
	Erster Treffer	Wer den ersten Frag einsackt
	Beeindruckend	Mit dem [#U-Mod] oder dem [#Kompressionsgewehr] zwei Treffer mit dem alternativen Modus (Sniper) hintereinander oder gleichzeitig.
	Exzellent	Zwei Frags innerhalb sehr kurzer Zeit nacheinander.
	Ass	5 Frags ohne selbst gefraggt zu werden.
	Experte	10 Frags ohne selbst gefraggt zu werden.
	Meister	15 Frags ohne selbst gefraggt zu werden.
	Champion	20 Frags ohne selbst gefraggt zu werden.
-	Verweigert	Ein Gegner greift sich ein Item *direkt* bevor man selbst es erreicht.

Nach dem Match

Symbol	Titel	Beschreibung
	Effizienz	50% Treffsicherheit während dem Match
	Scharfschütze	Mindestens fünf mal „Exzellent“.
	Weiße Weste	Kein einziges mal gefraggt werden.

Symbol	Titel	Beschreibung
	Logistiker	Sehr viele Items einsammeln
	Taktiker	Mit jeder verfügbaren Waffe mindestens einen Frag erzielen.
	Sprengstoffexperte	Viele Frags mit explosiven Waffen.

Nach CTF-Matches

Symbol	Titel	Beschreibung
	MVP („Most valuable Player“)	Der Spieler, der dem Team die meisten Punkte eingebracht hat.
	Verteidiger	Derjenige, der am häufigsten den Gegner mit der eigenen Flagge abgeschossen hat.
	Abfänger (Audio: Interceptor)	Der Spieler, der die eigene Flagge am häufigsten zurück gebracht hat.
	Krieger	Der Spieler mit den meisten Frags.
	Träger	Derjenige, der am häufigsten die Flagge abgeliefert hat.
	Tapferkeit	Der Spieler, der am häufigsten gefraggt wurde.

Section 31



Es gibt, ausgehend von vorhandenen Grafiken und Sounds, noch eine weitere Empfehlung, nämlich „Section 31“. Diese ist aber im Code so angelegt, dass sie nie angezeigt werden kann. Ausgehend von den auskommentierten Zeilen dort (Funktion `CalculateSection31Award`) dürfte es aber folgendes zugleich voraussetzen:

- Mindestens 75% Treffsicherheit,
- weiße Weste,
- Champion und
- Scharfschütze.

Maps, Mods und Models

Wie für alle beliebten Spiele, die es ermöglichen, gibt es für EF auch von Privatpersonen erstellte Maps, Mods, oder Models (Spieler-Figuren, aka Skins). Grundsätzlich gilt für alle drei dasselbe: Vom Anbieter runterladen, *Readme lesen* und entsprechend handeln. Wenn keine Readme da ist oder keine Installations-Hinweise gegeben sind, die `*.pk3`-Dateien ins Verzeichnis `baseEF` der EF-Installation kopieren. Alle Maps und Modells, aber auch die meisten Mods sind als `*.pk3`-Datei erstellt. Eine `*.pk3`-Datei ist im Großen und Ganzen nichts anderes als eine umbenannte `*.zip`-Datei, in der alle benötigten Daten sind, damit die Map, das Model, oder der Mod läuft. Einige wenige Maps benötigen aber das Expansion-Pack für EF, weil sonst benötigte Texturen fehlen, bzw. die Map sonst nicht geht. Nach dem man die Datei reinkopiert habt, sollten die Maps bzw. Modells direkt im Spiel auswählbar sein. Ein paar Maps bilden da eine Ausnahme, denn sie sind nur mit dem Befehl `/map <mapname>` erreichbar (nicht im Auswahl-Menü). `<mapname>` ist natürlich der Name der Map die ihr laden wollt. (Das liegt daran, dass keine Mapbeschreibung enthalten ist. Das ist eine `*.arena`-Datei im Ordner `scripts`, die u. A. beschreibt, welche Spieltypen damit spielbar sind.) Ob Mods verwendet werden, erkennt man häufig erst im Spiel, wenn Waffen-Skins verändert sind, oder ähnliches auftritt.

Mods

Mods können das Spielerlebnis teils erheblich bereichern, z. B. der Pinball-Mod (der Gegner muss aus dem Spielfeld geschossen werden, statt gefraggt). Manche haben aber das Problem, dass dann das Original-Spiel nicht mehr spielbar ist. Manche sind auch von Haus aus eingebaut, z. B. der Desintegration-Mod (nur Kompressionsgewehr, immer auf alternativem Feuermodus, dafür unbegrenze Munition). Hier mal ein paar Videos zu Mods für EF 1:

- Let's Play Star Trek: Enterprise - Regeneration
- Asteria Update and Video Testing
- RPG-X Station Asteria - Shuttlepod Full Launch

Und noch ein paar tatsächliche Mods:

- Teamelite, ein Specialist-Mod im Stile von Team Fortress für EF 1
- Star Trek: Enterprise NX01 Demo, ein (unfertiger) Mod, der auf der Enterprise NX-01 spielt
- Nebula 2 - Revenge of the Borg
- Star Trek Voyager Game Project (neues EF auf Basis Unreal Engine 4, hammer Grafik im Vergleich)
- Elite Force: Graphic Overhaul Project, EF mit neuen Grafiken versehen

Einige Mods werden auch heute noch für laufende Server dauerhaft verwendet. Darunter finden sich:

- EFAdmin
- Gladiator Arena
- SuperFFA
- TOS Weapons II (tw2)
- vv-CTF
- Power Weapons (Pwrweapon)

pak92.pk3

Der eine oder andere wird sich vielleicht schon mal gewundert haben, warum ihm Version 1.37 angezeigt wird, obwohl er 100%ig sicher war, dass er die original Binary (ggf. gecrackt) gestartet hat, die eben Version 1.2 wäre. Unter diesen Umständen macht es Sinn, mal im `baseEF`-Ordner nachzusehen, ob dort die Datei `pak92.pk3` liegt. Diese hat nämlich auch diese Auswirkung. Sie ist Teil der Arbeit von Thilo. Es gibt eine Unterscheidung, ob etwas in den Binaries (bzw. Libraries) erledigt wird, oder über vorkompilierte QVMs (eine Art Sandboxing-Methode von Quake 3, um Code portabel zu compilieren). Teile der Funktionalität sind in den QVMs realisiert, die in `pk3`-Dateien ausgeliefert werden. Auch ohne seine Binary-Version 1.37 hat man dann eben eine Anzeige als Version 1.37, durch das `pk3`-Archiv. Es ist aber offensichtlich etwas verwirrend, dass die angezeigte Version dann nicht mehr mit der Version der Binary übereinstimmen muss. Einige Dinge, die (wahrscheinlich, ausgehend von den diffs von Thilo) verändert wurden:

- Verbesserte Zufallsfunktionen in `q_math`,
- Veränderung der Schild-Dicke (hier gab es einen Bug, der dazu führt, dass die Dicke abhängig war, auf welche Weise das Schild aktiviert wurde),
- Verbot von Selbstmorden im Assimilation-Modus (richtig angewandt verhinderte es sonst, dass man einen Federation-Spieler assimiliert, weil der sich vorher noch schnell selbst umbringen konnte, und damit wieder spawnen),
- Ausschluss verbotener Zeichen im Nickname,
- `localhost` braucht zum Beitreten kein Passwort mehr und
- Client-seitige Ignore-Funktionen.

Models

Models sind im Prinzip Skins für die Spielfigur, Waffen oder auch neue Fadenkreuze. Models haben aber einen Pferdefuß: Wenn der Server sie nicht ebenfalls hat, wird clientseitig umgestellt auf das Standard-Model. Man wird also (nach einigen Tests mit EF 1.2) nicht unsichtbar oder ähnliches. Hat der Server das Model ebenfalls, wird es von den Clients ohnehin runter geladen und steht damit dann allen zur Verfügung.

Maps

Maps sind am wenigsten problematisch. Es ist nur wichtig, dass alle dieselbe Version spielen und ggf. weitere von einer Map benötigte `*.pk3`-Dateien vorhanden sind. Meistens findet man aber sowieso nur die neueste Version einer Map, oder die älteren sind als solche gekennzeichnet. Wichtig bei der Auswahl der Maps für einen dedicated-Server ist nicht zuletzt, dass möglichst viele Leute die Maps haben (bzw. nicht all zu viele custom Maps benötigt werden), oder sie sich schnell beschaffen können (z. B. Mappack auf einen FTP legen, denn der Standard-Download über EF dauert einfach echt lange, siehe auch Map-Downloads beschleunigen) und dass die Maps nicht all zu rechenintensiv sind. Gerade auf älteren Rechern können überdimensionierte oder überdetailierte Maps ruckeln, was vor allem bei einem schnellen Spiel wie EF eher ungünstig ist. Auf aktuellen Systemen fällt das aber längst nicht mehr ins Gewicht.

Angenehm ist, wenn eine Map nicht zu lange gespielt wird. Jeder Spieler hat so die eine oder andere Map, auf der er/sie auf keinen grünen Zweig kommt. Daher sollten die Limits (Zeit, Capture, Kills) sinnvoll gewählt sein. Grobe Richtwerte sind 30 Kills/5 Captures, 15 Minuten Zeitlimit. Dann gibt es

auch sowas wie 100 Kills 10 Death oder so nicht. Auf manchen Karten ist z. B. der mit dem Bogenschweißer einfach nahezu unschlagbar. Da macht es dann auch keinen Spaß, 100 Kills am eigenen Leib mit zu erleben. Ebenfalls sollte es eine recht abwechslungsreiche Auswahl an Maps sein. Zuerst CTF und danach DM auf der selben Karte ist langweilig.

Die Eignung für bestimmte Modi oder Spieltypen sollte auch berücksichtigt werden. Desintegration z. B. spielt sich auf größeren Maps besser, als auf kleineren. Ebenfalls sollte die Anzahl der Spieler berücksichtigt werden. Auf einer riesigen Map wenige Spieler macht keinen Spaß, eine knallvolle Map ebenfalls nicht. Ersters lässt sich ggf. durch Auffüllen mit Bots ausgleichen.

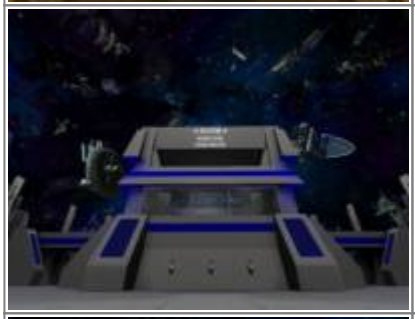
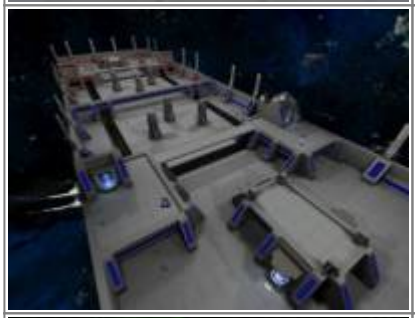
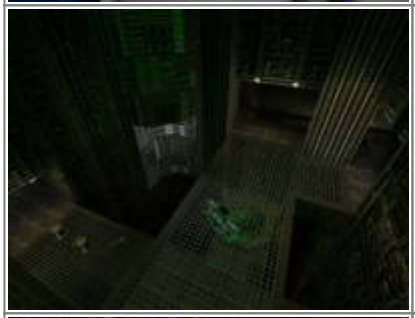
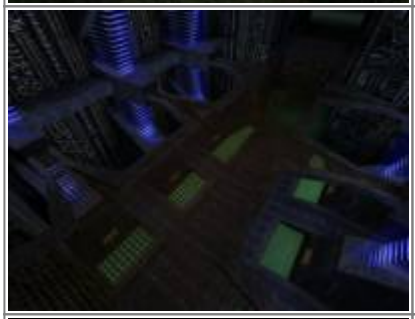
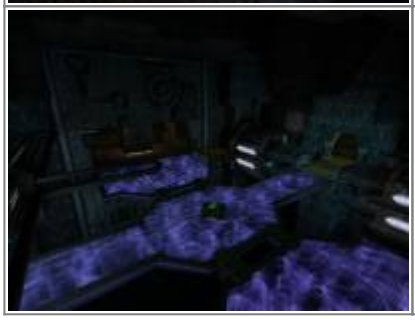
Der Befehl zum Aufrufen von Maps (`/map <mapname>`) ist auf *alle* Maps anwendbar, aber nur bei wenigen wirklich notwendig. Wichtig ist hier aber, dass nicht der Name des `*.pk3` gemeint ist, sondern der Name der `*.bsp`-Datei im Ordner `maps` der `*.pk3`-Datei. Mit diesem Namen werden Maps übrigens auch für Map-Cycles eines Dedicated Server aufgerufen. Das heißt aber, dass die `*.pk3`-Datei auch ganz anders heißen kann, als die eigentliche Map (z. B. enthält `pak0.pk3` aus der Basis-Installation einige Maps, nicht nur eine). Im Zweifelsfalle gilt also nachsehen. Es hilft dann, die `*.pk3` so zu benennen, wie die `*.bsp` heißt. (Sofern sie nicht mehrere Maps enthält.) Die Groß- und Kleinschreibung ist bei Aufruf dieses Befehls dann egal.

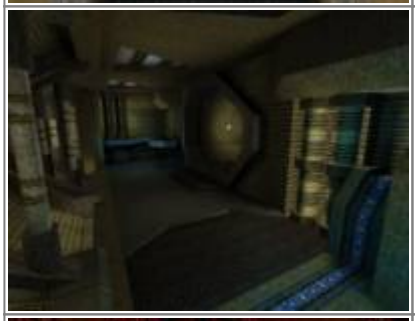
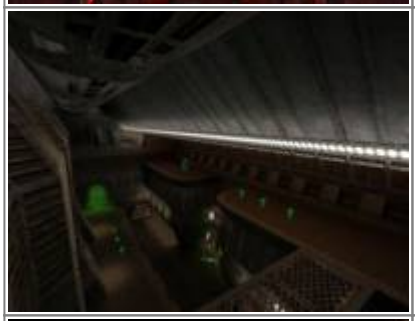
Ggf. wichtig ist auch noch, ob eine Map mit Bots gespielt werden kann (sprich, Waypoints hat). Das erkennt man daran, ob im Ordner `maps` des `*.pk3` auch eine `*.aas`-Datei mit dem Namen der Map vorliegt. Ist dies nicht der Fall, kann auf der Map kein Bot spielen.

Standard Maps

Wie in den allermeisten Spielen ist auch in Eliteforce ein Satz Maps mitgeliefert, die jeder, der das Spiel sauber installiert hat, haben muss. Im weiteren Sinne kann es noch zum Bereich Standard-Maps zählen, wenn man die Maps des Expansion Pack mit rein nimmt. Die haben viele, aber nicht zwingend jeder. Hier eine Liste der Maps, die von Haus aus in EF (im `pak0.pk3`) enthalten sind:

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_and1	CTF, TDM, FFA	10	-
	ctf_kln1	CTF, TDM, FFA	10	-

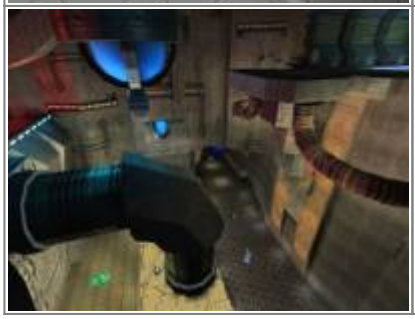
Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_kln2	CTF, TDM, FFA	12	-
	ctf_voy1	CTF, TDM, FFA	8	10
	ctf_voy2	CTF, TDM, FFA	8	15
	hm_borg1	TDM, FFA	6	10
	hm_borg2	TDM, FFA	8	20
	hm_borg3	TDM, FFA	12	25

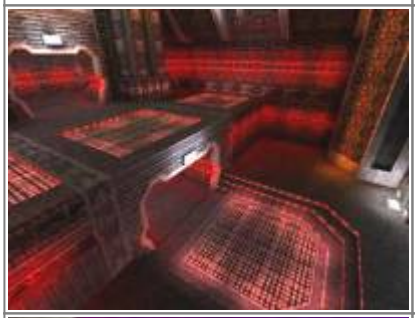
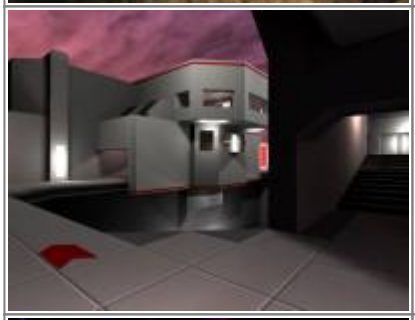
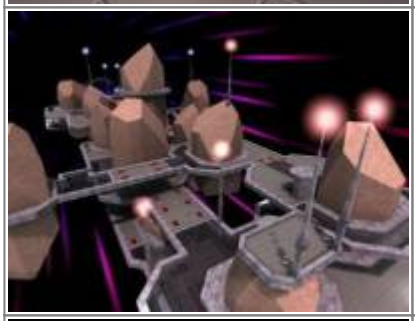
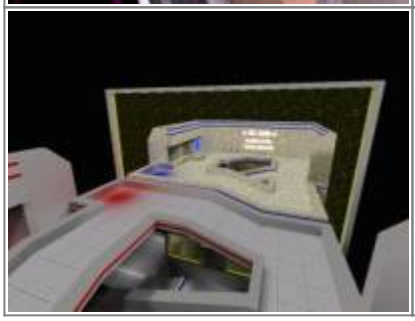
Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	hm_cam	TDM, FFA	6	20
	hm_dn1	TDM, FFA	12	15
	hm_dn2	TDM, FFA	16	20
	hm_for1	TDM, FFA	8	10
	hm_kln1	TDM, FFA	6	10
	hm_noon	TDM, FFA	4	10

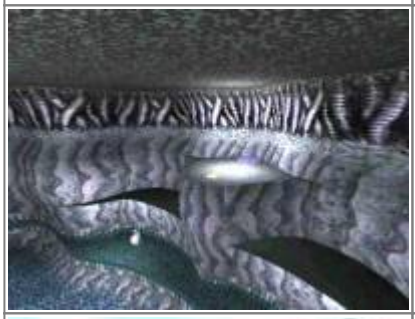
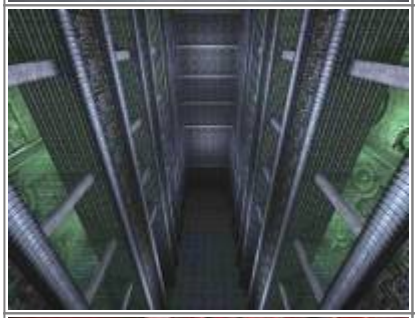
Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	hm_scav1	TDM, FFA	8	15
	hm_voy1	TDM, FFA	12	15
	hm_voy2	TDM, FFA	8	10

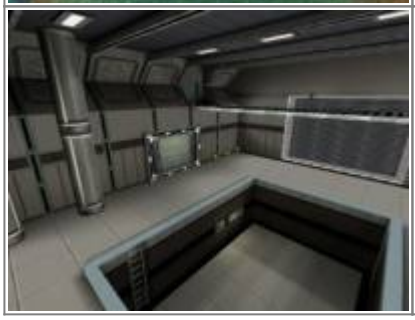
Expansion Pack Maps

Die folgenden Maps sind regulär im Expansion Pack (im `pak3.pk3`) enthalten:

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_breach	CTF, TDM, FFA	12	20
	ctf_dn1	CTF, TDM, FFA	8	15

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_for1	CTF, TDM, FFA	12	15
	ctf_neptune	CTF, TDM, FFA	6	10
	ctf_oldwest	CTF, TDM, FFA	6	15
	ctf_reservoir	CTF, TDM, FFA	8	15
	ctf_singularity	CTF, TDM, FFA	8	20
	ctf_spyglass2	CTF, TDM, FFA	8	10

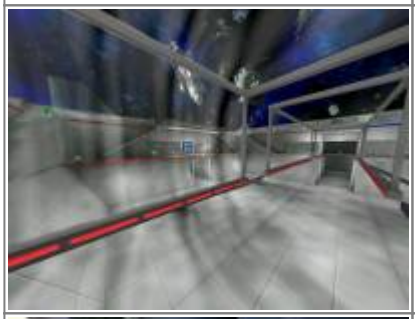
Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_stasis	CTF, TDM, FFA	4	10
	hm_altar	TDM, FFA	8	10
	hm_blastradius	TDM, FFA	8	15
	hm_borgattack	TDM, FFA	8	20
	hm_for2	TDM, FFA	10	15
	hm_raven	TDM, FFA	4	10

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	hm_temple	TDM, FFA	8	25
	hm_voy3	TDM, FFA	6	20

Elite Force Player Maps

Es gibt noch ein paar zusätzliche Maps, die auf der CD des Expansion Packs drauf sind. Sie werden als „Elite Force Player Maps“ bezeichnet, und befinden sich als Installer `EliteForcePlayerMapsGM.exe` im Verzeichnis `Setup/PlayerMaps`. Dieser Installer platziert letztlich nur die Datei `playermaps.pk3` im `BaseEF` Order. Dieses pk3 enthält die folgenden Maps:

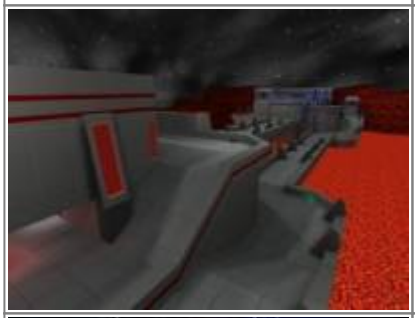
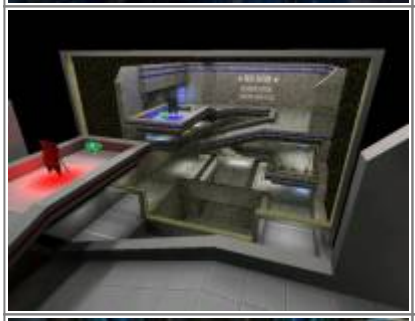
Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	borg210	TDM, FFA	8	20
	ctf_ambushv2	CTF, TDM, FFA	8	25

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_thebunkerv2	CTF, TDM, FFA	6	20
	hm_209	TDM, FFA	-	30
	shipwrecked	TDM, FFA	8	10

Bonus Holomatch Map Pack

Es gibt noch eine handvoll weitere Maps, die von Raven Software erstellt wurden, aber nicht auf den CDs ausgeliefert wurden. Diese waren nur über das Internet, als Mappack bzw. einzelne Maps verfügbar. Sie liegen als Dateien `ctf_geothermal.pk3`, `ctf_spyglass.pk3` und `stvef_mappack.zip` vor. (Zu finden z. B. hier und hier.) In `stvef_mappack.zip` findet sich die eigentliche Map-Datei `raven_trekmaps1.pk3`. Sie enthält die drei Maps `ctf_canyon`, `ctf_space` und `hm_dn3`.

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_canyon	CTF, TDM, FFA	10	20

Level-Screenshot	Mapname	Spielmodi	Empfohlene Anzahl Spieler	empf. Fraglimit
	ctf_geothermal	CTF, TDM, FFA	10	20
	ctf_space	CTF, TDM, FFA	6	20
	ctf_spyglass	CTF, TDM, FFA	6	20
	hm_dn3	TDM, FFA	6	20

Maperstellung

Über die eigentliche Erstellung der Maps möge bitte jemand anders etwas schreiben. Trotzdem einige der Informationen, die mir über den Weg gelaufen sind. Zunächst mal zum Format der Map-Dateien: Diese sind wie oben angesprochen im Prinzip `*.zip`-Dateien, die Texturen und Mapgeometrie sowie Bot-Daten und ggf. zusätzliche Sounds enthalten.

Bots

Maps haben nicht von Haus aus Bot-Unterstützung. Ob diese vorliegt, kann man leicht herausfinden, wenn man ins `*.pk3`, in den Ordner `maps` sieht. (Ist dieser nicht vorhanden, ist das `*.pk3` keine Map-Datei, sondern entweder eine Unterstützungsdatei (mit z. B. zusätzlichen Texturen oder Sounds) oder aber ein Model oder Bot.) Liegt darin neben der eigentlichen Map (`*.bsp`) eine `*.aas`-Datei mit demselben Namen vor, hat sich jemand die Mühe gemacht und der Map Botunterstützung gegeben.

Diese lässt sich im Prinzip auch nachrüsten. Dazu muss man die `*.bsp`-Datei aus dem `*.pk3` mit BSPC in ein `*.aas` konvertieren. Eine Befehlszeile dazu sieht wie folgt aus:

```
bspc -bsp2aas <pk3-Datei>/maps/bspfile.bsp
```

Die daraus erhaltene Datei kann man dann wieder ins `*.pk3` in den Ordner `maps` kopieren. Es gibt noch eine Reihe anderer Funktionen von BSPC, die hier noch nicht erklärt sind. Für Windows steht mit dem BSPC-GUI auch eine Arbeitserleichterung bereit. Meistens ist es aber so, dass auf Maps, die keine Bot-Unterstützung haben, nicht einfach nur der Mapper zu faul dazu war, sondern es technische Probleme gibt, die Waypoints nachzurüsten. Versuchen kann man es aber trotzdem mal, wenn man sich nicht sicher ist.

Abkürzungen

Die folgenden Abkürzungen sind sicher noch interessant:

- CTF = Capture the Flag
- FFA = Free For All (=Deathmatch)
- TDM = Team Deathmach
- PIN = Pinball
- DESI = Desintegration (alle haben nur Sniper-Rifle, die auch nur im Sniper-Modus feuern können)

Screenshots

Nachdem mancher auf LANs gerne Statistiken führt, oder sich einfach gerne hinterher ein paar Screenshots anschauen will, gibt es die Funktion entweder als Menü-Punkt wenn man während dem Spiels Escape drückt, oder aber viel einfacher mit der F11-Taste. Ich empfehle das immer am Ende einer Map zu machen, wenn die Liste mit den Punkten rein flattert. Ist häufig am aussagekräftigsten wenn man sich das nochmal genauer anschaut.

Demos

Demos sind sozusagen die Life-Variante der Stats. Auch in EF gibt es die Möglichkeit Demos zu erstellen und anzusehen. Die Demos erstellt man mit den folgenden Konsolen-Befehlen (die Konsole erreicht ihr mit der ^-Taste links oben auf eurer Tastatur bzw. mit ~ bei Thilos Binaries):

- Vorbereitung: `/g_synchronousClients 1`
- Demo Demoname aufnehmen: `/record Demoname`
- Synchronisierung abschalten: `/g_synchronousClients 0` (sonst ruckelt es im laufenden Spiel)
- Aufnahme beenden: `/stoprecord`
- Demo `demoname` ansehen: `/demo demoname`

Die gespeicherten Demos findet man im Verzeichnis `baseEF\Demos` der EF-Installation. Man kann sie an andere Spieler weiter geben. Übrigens gibt es zum Ansehen von Demos auch die wesentlich

bequemere Variante mittels des `Demos`-Befehls im Hauptmenü vom Holomatch. Einfach auswählen und ab geht's.

Cheats

Die folgenden Cheats gehen *nur* im Singleplayer (nein, auch nicht, wenn man im Multiplayer nur Bots hinzufügt). Wenn man sie im Multiplayer eingibt, gehen sie als Chatmessages raus, was entsprechende Reaktionen bei den Mitspielern verursachen dürfte... Die Cheats lassen sich nur nutzen, wenn man sie aktiviert. Entweder man hat in der Singleplayer-Config bereits `sv_cheats 1` stehen, oder aber, man gibt es in die Konsole ein. Danach gehen die folgenden Codes:

Cheatliste

Cheat	**Wirkung**
`give #`	Gegenstand mit Nummer # geben (siehe Liste)
`give weapons`	Alle Waffen verfügbar
`give health #`	Setzt die Gesundheit auf den Wert # (1-200)
`give armor #`	Setzt die Rüstung auf den Wert # (0-200)
`give weaponnum #`	Gibt die Waffe mit der Nummer # (siehe Liste)
`name #`	Setzt den eigenen Nick auf #
`model #`	Setzt das eigene Model auf #
`setviewpos X Y Z (R)`	Setzt einen an Position X,Y,Z, mit dem Drehwinkel R
`ui_closingcredits`	Abspann anzeigen
`dir video`	Zeigt die Liste der verfügbaren Videos an
`cinematic <movie>`	Spiel das Video `<movie>` ab
`give tricorder`	Gibt den Tricorder
`undying`	Gesundheit und Rüstung werden auf 999 gesetzt
`noclip`	Man kann fliegen und durch Wände gehen
`notarget`	Gegnerische Einheiten sehen einen nicht mehr als Ziel an
`map_brig`	Verstecktes Level
`cg_thirdperson 1`	Third-Person-Ansicht an (0 für aus)
`cg_thirdpersonrange #`	Abstand zur Spielfigur einstellen
`devmap #`	Cheats auf einem Server für die Map # aktivieren
`map #`	Startet die Map #
`kill`	Selbstmord
`quit`	Spiel beenden
`god`	God-Mode

Gegenstände

Nummer	**Gegenstand**
1	Phaser
2	Compression Rifle
3	Scavenger Rifle
4	Tetryon Disruptor

Nummer	Gegenstand
5	IMOD
6	Tricorder
7	Health
8	Ammo
9	Weapons
10	All
11	Stasis Weapon
12	Grenade Launcher
13	Photon Burst
14	Dreadnought Weapon
15	Armor
16	Paladin Weapon
17	Desperado Weapon
18	Klingon Blade
19	Bot welder
20	Bot laser
21	Bot rocket
22	Forge proj
23	Forge psych
24	Parasite
25	Borg weapon
26	borg taser
27	Paladin weapon
28	Alle

Config-Parameter

Diese Liste enthält häufig genutzte Config-Parameter. Sie ist allerdings sicher nicht abschließend.

Variable	Bedeutung/Effekt
`com_zoneMegs "x"`	Vermutlich(!) der Arbeitsspeicher in MB, den EF maximal für Items nutzen darf.
`com_soundMegs "x"`	Arbeitsspeicher in MB, den EF maximal für Sounds nutzen darf.
`com_hunkMegs "x"`	Arbeitsspeicher in MB, den sich EF maximal genehmigen darf. Systemlimits wie die 2 GB-Grenze unter XP können damit nicht überschritten werden!
`m_filter "x"`	Mausfilter, wenn auf 1 gesetzt, statt auf 0 (standard), werden die Mausbewegungen etwas gleichmäßiger
`in_joystick "x"`	Soll ein Gamepad/Joystick verwendet werden? (1=ja, 0=nein)
`in_mouse "x"`	Soll eine Maus als Eingabegerät verwendet werden? (1=ja, 0=nein)
`g_subtitles "x"`	Sollen Untertitel angezeigt werden? (2=in Videoszenen, 1=ja, 0=nein)

Variable	Bedeutung/Effekt
r_finish "x"	Sollte es aufgrund von Problemen mit den Eingabegeräten zu Laggs kommen, auf 1 setzen. Dadurch werden die FPS etwas weniger, aber dafür sind die Timings wieder in Ordnung
r_ignorehwgamma "x"	Wenn auf 1 gesetzt, kontrolliert EF selbst die Gamma-Werte (bzw. man kann sie vom Spiel aus einstellen), ansonsten die Grafikkarte (ist unter XP sehr nützlich, da EF die Einstellungen der Helligkeit oft ignoriert, sodass man es besser auf 0 setzt)
cg_autoswitch "x"	Soll automatisch auf eben eingesammelte Waffen gewechselt werden? (1=ja, 0=nein)
cg_drawAmmoWarning "x"	Soll bei niedrigem Munitionsstand eine Warnung zu sehen sein? (1=ja, 0=nein)
cg_draw2D "x"	Sollen statt der 3D-gerenderten Bilder für Items die 2D-Bilder verwendet werden? (1=ja, 0=nein), betrifft auch das HUD
cg_drawCrosshair "x"	Welches Fadenkreuz angezeigt werden? (0=keines, sonst durchnummeriert bis 10)
cg_crosshairSize "x"	Größe des Fadenkreuzes, 24 ist default
cg_drawCrosshairNames "x"	Sollen die Spielernamen am Fadenkreuz angezeigt werden, wenn beide sich nah sind? (1=ja, 0=nein)
cg_drawFPS "x"	Soll die momentane FPS-Rate angezeigt werden? (1=ja, 0=nein)
cg_drawGun "x"	Soll das Model der momentan verwendeten Waffe gezeichnet werden? (1=ja, 0=nein) Falls nein, hat man zwar etwas mehr Übersicht und die FPS-Rate steigt ein wenig, aber welche Waffe man gerade verwendet, muss man sich anderweitig erschließen.
cg_forceModel "x"	Werden im Multiplayer für alle Spieler dieselben Models verwendet? (1=ja, 0=nein) Hat vor allem dann Vorteile, wenn Leute mit Models rum laufen, die man selbst nicht hat. Die erscheinen sonst nämlich unsichtbar. Hat aber den Nachteil, dass man einzelne Spieler nicht mehr vom Aussehen her voneinander unterscheiden kann.
cg_marks "x"	Sollen Trefferspuren an Wänden verbleiben? (1=ja, 0=nein)
cg_shadows "x"	Sollen Schatten angezeigt werden? (2=volumetrische, 1=kugelfömige, 0=keine), siehe auch r_stencilbits, Modus 2 muss von der Grafikkarte unterstützt sein
cg_simpleItems "x"	Sollen Items in 3D angezeigt werden (1=nein, 0=ja)? 2D sieht schlechter aus, erhöht aber die FPS
com_FirstTime "x"	Ist das Intro noch nicht abgespielt worden? (1=ja, 0=nein, falls nein, wird es nach dem Start des Spiels erst ein mal abgespielt)
cg_bobup "0.00x"	Legt fest, wie stark der Spieler beim Gehen „pendelt“, 0 schaltet das Pendeln komplett ab.
cg_bobpitch "0.00x"	Legt fest, wie stark der Spieler beim Gehen „schwankt“, 0 schaltet das Schwanken komplett ab.

Variable	Bedeutung/Effekt
cg_bobroll "0.00x"	Legt fest, wie stark der Spieler beim Gehen „vorwärts/rückwärts wackelt“, 0 schaltet die Wackelei komplett ab.
cg_runpitch "0.00x"	Legt fest, wie stark der Spieler beim Rennen „schwankt“, 0 schaltet das Schwanken komplett ab.
cg_runroll "0.00x"	Legt fest, wie stark der Spieler beim Rennen „vorwärts/rückwärts wackelt“, 0 schaltet die Wackelei komplett ab
r_allowSoftwareGL "x"	Soll das Spiel software-gerendert werden? (1=ja, 0=nein), macht nur Sinn, wenn man tatsächlich Schwierigkeiten beim normalen Rendern hat
r_clear "x"	Sollen Lücken zwischen Texturen gefüllt werden? (1=ja, 0=nein) Sieht zwar besser aus, frisst aber mehr Rechenleistung.
r_colorbits "x"	Farbtiefe, 16 oder 32 (bit), siehe auch r_depthbits
r_depthbits "x"	Ähnlich wie r_colorbits, nur bei 0 Bit wird die Einstellung für den Windows-Desktop übernommen.
r_detailtextures "x"	Sollen verbesserte Texturen verwendet werden? (1=ja, 0=nein)
r_displayRefresh "x"	Bildschirmwiederholrate, sollte auf denselben Wert gesetzt werden, wie der Monitor maximal ausgibt. Auf modernen LCD-Monitoren ist das meisten 60, bzw. bei 3D-fähigen, 120
r_drawSun "x"	Soll Sonnenlicht gerendert werden? 1=ja, 0=nein, Betrifft nur Karten, auf denen Sonnenlicht vorkommt.
r_dynamiclight "x"	Sollen dynamische Lichter gerendert werden (z. B. Lichtschimmer von Explosionen)? 1=ja, 0=nein
r_ext_compress_textures "x"	Sollen komprimierte Texturen verwendet werden? 1=ja, 0=nein, ausgeschaltet sieht besser aus, belastet aber ggf. das System deutlich stärker
r_ext_multitexture "x"	Sollen multi-texturing verwendet werden (mehrere Ebenen an Texturen gleichzeitig)? 1=ja, 0=nein
r_ext_texture_filter_anisotropic "x"	Sollen anisotrope Filter angewendet werden? 1=ja, 0=nein, Sollte man aus optischen Gründen verwenden, alles was weiter weg ist, wird dann schärfer dargestellt
r_fastsky "x"	Soll statt der optisch ansprechenderen Variante Himmel und Portale nicht gerendert werden? 1=ja, 0=nein
r_gamma "x.y"	Gamma-Helligkeitswert, der Defaultwert ist 1.0
r_lodbias "x"	Model-Detail-Level, Werte zwischen 0 und 2 sind erlaubt, mit 2 am wenigsten detailliert, aber höchste FPS-Rate.
r_nocull "x"	Soll alles gerendert werden, oder nur sichtbare Teile? 0=alle, 1=nur sichtbare
r_nocurves "x"	Sollen Kurven gerendert werden? 1=ja, 0=nein, hat einen starken Effekt auf Optik und Leistung
r_norefresh "x"	Sollen sämtliche Bildschirmaktualisierungen unterlassen werden? 1=ja, 0=nein, sollte auf bleiben, da sonst nichts mehr passiert.

Variable	Bedeutung/Effekt
r_overBrightBits "x"	Helligkeit insgesamt, 0=keine zusätzliche Aufhellung, 1 ist normal,
r_mapoverbrightbits	Helligkeit der Maps, 0=keine zusätzliche Aufhellung, 1 ist normal
r_picmip "x"	Mipmap Level, Werte von 0 bis 4 sind erlaubt, beeinflusst die Bildqualität, 4=schnellste, 0=optisch beste, aber langsamste Einstellung
r_smp "x"	Sollen mehrere Prozessoren verwendet werden? 1=ja, 0=nein
r_stencilbits "x"	Schablonen-Puffer, wirkt sich u. A. auf volumetrische Schatten (cg_shadows) aus, sollte bei Benutzung auf 8 gesetzt werden.
r_subdivisions "x"	Komplexität von Rundungen, 1 ist der optisch beste, aber langsamste Wert, 999 ist der schnellste, aber optisch kantigste Wert
r_swapInterval "x"	Vsync, sollen Game-FPS und tatsächliche (vertikale) Wiederholfrequenz angeglichen werden? 1=ja, 0=nein, 0 macht eigentlich nur für Benchmarks sind, da mehr als die physikalische Wiederholfrequenz ohnehin nicht angezeigt wird
r_texturebits "x"	Sollen 16 oder 32 Bit Texturen verwendet werden?
r_textureMode "x"	Texturmodus, kann die Werte GL_NEAREST_MIPMAP_NEAREST (schlechteste), GL_LINEAR_MIPMAP_NEAREST (linear, default), GL_NEAREST_MIPMAP_LINEAR (bilinear=zweitbeste) oder GL_LINEAR_MIPMAP_LINEAR (trilinear=beste) haben
s_a3d_weapons "x"	A3D-Waffen-Sounds (z. B. Wiederhall) an? 1=ja, 0=nein
s_enableA3D "x"	A3D an? 1=ja, 0=nein
s_khz "x"	Samplerate, 22 oder 11 (kHz)
s_loadas8bit "x"	Sollen 8bit Samples verwendet werden? 1=ja, 0=nein, sollte üblicherweise aus sein, auch wenn es geringfügig Systemleistung einspart
s_volume "x"	Lautstärke von 0 bis 1
s_musicvolume "x"	Musiklautstärke von 0 bis 1
s_volumeVoice "x"	Voiceover-Lautstärke von 0 bis 1
s_doppler "x"	Doppler-Effekt, von 0 bis 10, sollte man normalerweise auf dem defaultwert lassen (=nicht in der Konfig stehen oder auskommentiert haben)
s_reflect "x"	Audioreflektionen an? 1=ja, 0=nein
s_occlude "x"	Dämpfung/Umgebungseinflüsse auf den Sound an? 1=ja, 0=nein
s_geometry "x"	A3D-Nachbearbeitung für Geometrieeinflüsse an? 1=ja, 0=nein
s_geom_reverb "x"	Geometriebeeinflusster Hall an? 1=ja, 0=nein
s_mixahead "x.y"	Vorausberechnung von Sound, je höher, desto mehr Systembelastung

Variable	Bedeutung/Effekt
s_numpolys "x"	Anzahl maximal zur Berechnung verwendeter A3D-Polygone, je höher, desto bessere 3D-Sounds, aber umso mehr Systembelastung, 0=aus
s_initsound "x"	Soll überhaupt Sound verwendet werden? 1=ja, 0=nein (ist nicht dasselbe wie Lautstärke auf 0!)
s_2dvolume "x"	Lautstärke der 2D-Sound von 0 bis 1
cg_lagometer "x"	Soll das Lagometer/der Netgraph angezeigt werden? 1=ja, 0=nein
snaps "x"	World Updates zum Client pro Sekunde gesendet, 30 ist ein guter Start-Wert, muss aber ggf. abhängig von der eigenen Verbindung angepasst werden.
rate "x"	Datenrate in Bytes/sek., hängt von der eigenen Verbindungsrate ab, kann bei zu hohen Pings helfen, wenn man den Wert reduziert.
com_maxfps "x"	Maximale Framerate, sollte man eher nicht verwenden, wenn, dann eher cl_maxpackets
cl_maxpackets "x"	Pakete pro Sekunde zum Server gesendet, 30 ist für V90 Modem, für DSL bzw. Standleitungen kann man hier sicherlich weiter rauf gehen
cg_deferPlayers "x"	Verzögerung von Model-Laden? 1=ja, 0=nein, verzögert das Nachladen von Models beim Beitritt anderer Spieler, damit es nicht gerade in genau dem Moment laggt. Nachgeladen wird dann erst beim nächsten Spawning.
cl_timenudge "x"	Ping-Lagg simulieren, negative Werte können beim Onlinespiel Probleme verrringern (-½·durchschnittlichen Ping verwenden)
cl_packetdup "x"	Paket-Doppelung, hilfreich bei Loss, da Pakete mehrfach versendet werden
cg_predictItems "x"	Item-Voraussagen an? 1=ja, 0=nein, sollte bei guter Anbindung auf 0 belassen werden, da es auf 1 manchmal fälschlicherweise aufgesammelte Items beim Client ergibt, die gar nicht aufgenommen wurden
cl_nodelta "x"	Delta-Kompression deaktiveren? 1=ja, 0=nein, sollte man grundsätzlich auf 0 belassen
net_noipx "x"	Soll IPX deaktiviert sein? 1=ja, 0=nein, sollte 1 sein, wenn man kein IPX verwenden *kann* (z. B. Win Vista und neuer auf 64 Bit)
net_noudp "x"	Soll UPD deaktiviert werden? 1=ja, 0=nein, sollte eigentlich immer 0 sein, da über das Internet ohnehin kein IPX geht.
cl_allowDownload "x"	Sollen fehlende Maps automatisch heruntergeladen werden? 1=ja, 0=nein
name "x"	Name des Spielers
model "x"	Model/Skin des Spielers
cg_drawAttacker "x"	Soll nach dem Frag angezeigt werden, von wem man gefraggt wurde? 1=ja, 0=nein
cg_drawIcons "x"	Sollen Icons für Rüstung, Gesundheit und Munition im HUD angezeigt werden? 1=ja, 0=nein

Variable	Bedeutung/Effekt
cg_draw3dIcons "x"	3d-Anzeige von Items an? 1=ja, 0=nein
cg_gibs "x"	Splatter Effekte, 1=ja, 0=nein
com_blood "x"	Blut an? 1=ja, 0=nein
ui_team_friendly "x"	Friendly Fire an? 1=ja, 0=nein
ui_ctf_friendly "x"	Friendly Fire für CTF an? 1=ja, 0=nein
cg_drawtimer "x"	Timer anzeigen 1=ja, 0=nein
cg_drawTeamOverlay "x"	Soll ein Team-Overlay angezeigt werden? 1=ja, 0=nein

Kommandos im Spiel

Befehl	Wirkung
set xyz blah	Setzt die Variable xyz auf den Wert blah, kann auch andere Variablen enthalten, die dann ersatzweise dafür eingesetzt werden z. B. set a blahfasel und set b "a blubber"
set g_synchronousClients 1	Synchronisiere Client und Server stetig
/record demoname	Beginnt eine Demo-Aufnahme mit dem Namen demoname auf
/disconnect	Verbindung zum Server unterbrechen
/demo demoname	Spielt die Demo mit dem Namen demoname ab
/rconpassword "passwort"	Setzt *beim Client* das rcon-Passwort. Hier ist wichtig, dass man den / am Anfang nicht vergisst, da sonst das rcon-Passwort als Chatmessage für alle Mitspieler raus posaunt wird(!). Am besten legt man sich das auf eine der Tasten.
/rcon befehl	Setzt den Befehl befehl auf dem Server ab, wenn man sich vorher mit rconpassword authentifiziert hat.

Befehle

Befehle werden auf der Spiele-internen Konsole mit einem / voran eingegeben. Auf der Server-Konsole selbst (sofern man die Version mit GUI nimmt) braucht man ihn nicht.

Befehl	Effekt
cmdlist	Liste der verfügbaren Befehle ausgeben
cvarlist	Liste der verfügbaren cvars ausgeben
map xyz	Wechselt auf die Map mit dem Dateinamen xyz
serverinfo	Zeigt Servereinstellungen wie Protokoll, Name, Limits, usw. an
meminfo	Zeigt Speicherinformationen an
status	Anzeige der Spielerdaten, z. B. num, score, ping, name, address, port, rate
dumpuser	Zeigt Spielerinformation an, wie name, sex, skill usw.
kick ID	Spieler mit ID kicken
addip IP	Spieler mit IP blocken
removeip IP	Spieler mit IP unblocken
g_banIPs	momentan bebannte IPs anzeigen lassen
addbot BOTNAME	Bot mit Namen BOTNAME hinzufügen
exec CONFIG	Führt das Konfig-File CONFIG aus

Befehl	Effekt
killserver	Beendet die aktuelle Map (nicht den Server)
svsay NACHRICHT	Sende die Textzeile NACHRICHT an alle Spieler wie eine Consolen-Nachricht
reconnect	neu mit dem Server verbinden
password "xxx"	Passwort setzen, um z. B. einem Server beitreten zu können
vid_restart	Startet die Video-Ausgabe neu, nützlich wenn man eine Video-Einstellung verändert hat
bind taste "befehl1; befehl2;"	Automatische Ausführung von Befehlen und Skripten auf eine Taste legen
echo "text"	Gibt den Text text aus.
team x	Wechselt in Team x, mit s=Spectator, r=rot, b=blau
condump <filename>	Speichert die bisherigen Konsolen-Ausgaben in die angegebene Datei ab.
writeconfig <filename>	Schreibt die aktuellen Einstellungen in die angegebene Datei
dir <ordnername>	Zeigt die Dateien (inkl. in *.pk3-Dateien) im angegebenen Ordner an
path	Listet die aktuellen Suchpfade (auch *.pk3-Dateien) und die offenen Dateien auf
map_restart	Startet die aktuelle Karte neu
protocol	Zeigt die Protokollversion des Servers an

Variablen

Variablen werden üblicherweise mit set oder einem Derivat davon nach dem Prinzip set variable "wert" gesetzt.

set-Variante	Bedeutung
set	setzt eine cvar
seta	setzt eine cvar und markiert sie als Archiv (heißt, sie wird in die Konfig übernommen, in hmconfig.cfg in den jeweiligen Mod-Ordnern bzw. baseEF)
sets	setzt eine cvar markiert sie als Serverinformation
sett	setzt eine cvar markiert sie als Tool
setu	setzt eine cvar markiert sie als Userinformation

sv_ am Anfang steht üblicherweise für Server-Variablen, cl_ am Anfang steht üblicherweise für Client-Variablen. Hier ist eine Liste von Variablen mit ihren Bedeutungen:

Variable	Bedeutung
rconpassword	Setzt das Passwort für rcon-Befehle *serverseitig*
rconAddress	Variable für Adresse des fernzubedienenden Servers
sv_hostname	Name, der in der Serverliste und beim Beitreten angezeigt wird
sv_maxclients	Maximale Anzahl an zugelassenen Clients
sv_maxPing	höchster zugelassener Ping
sv_minPing	kleinster zugelassener Ping
sv_allowdownload	Sind Downloads zu den Clients erlaubt? 1=ja, 0=nein
sv_pure	Müssen die Dateien beim Client dieselbe Prüfsumme haben, wie auf dem Server? 1=ja, 0=nein

Variable	**Bedeutung**
sv_killserver	Sind Clients in der Lage, einen killserver-Befehl abzusetzen? 1=ja, 0=nein
sv_mapChecksum	Sollen Client- und Server-Files verglichen werden? 1=ja, 0=nein
sv_nopredict	Darf der Client Vorhersagen verwenden? 1=nein, 0=ja
sv_privateClients	Dürfen nur Clients mit Passwort beitreten? 1=ja, 0=nein
sv_privatePassword	Passwort für den Serverbeitritt
sv_floodProtection	Setzt einen Schutz gegen zu viele Chatmessages 0=aus, 1=an)
timescale	Setzt server timescale modifier
sv_reconnectlimit	Maximale Anzahl an Reconnects für Clients
sv_serverid	Setzt eine Server-ID (Nutzen ist unbekannt)
sv_showloss	Soll der Loss zum Server angezeigt werden? 1=ja, 0=nein
sv_timeout	Server-Timout-Zeit festlegen
sv_zombietime	Setzt die Zeit, nach der inaktive Clients (Zombies) gekickt werden.
g_needpass	Aktiviert oder Deaktiviert das Serverpasswort (0=aus, 1=an)
logfile	1 = Logfile wird angelegt, 0 = kein Logfile
g_log	Setzt Name der Logdatei (wird automatisch im Verzeichnis $Home/.stvef/baseEF bzw. baseEF des Spiels angelegt)
net_port	UDP-Port, auf dem der EF-Server erreichbar sein soll
g_motd	Message of the Day (hier soll angeblich auch eine Datei angebbar sein, das ist noch zu testen)
cl_motd	0 deaktiviert „Meldung des Tages“ Anzeige (wir beim Laden der Karte am Client angezeigt), default ist 1
g_allowvote	Sind Abstimmungen erlaubt? 1=ja, 0=nein
g_dmgmult	Schadensmultiplikator, default ist 1
g_gravity	Gravitation, default=800
g_speed	Laufgeschwindigkeit, default=300
g_forcerespawn	Sollen Spieler zwangsweise wieder in Spiel einsteigen, wenn sie gefraggt wurden? 1=ja, 0=nein
timelimit	Maximale Laufzeit einer Map
fraglimit	Fraglimit, bei dem die Map endet, gilt bei TDM für alle Spieler eines Teams gemeinsam, im Assimilation- und Elimination-Modus sowie CTF keine Auswirkung
capturelimit	Capturelimit, bei dem die Map endet
g_gametype	Spieltyp, 0=FFA, 1=1on1, 2=Singleplayer, 3=Team Holomatch, 4=CTF
g_pModActionHero	Modus ActionHero, einer ist der Hero, hat mehr Health, mehr Waffen, ist aber das bevorzugte einzige Ziel, weil sein Frag 5 Punkte zählt, und der Angreifer selbst zum Hero wird, 1=aktiviert, 0=deaktiviert.
g_pModAssimilation	Modus Assimilation, Sternenflotte kämpft gegen Borg. Entweder spielt man für die Sternenflotte, und versucht die Borgkönigin zu eliminieren, oder als Borgdrohne/Borgkönigin, und assimiliert das gegnerische Team, um so mehr Drohnen und schließlich die Runde zu gewinnen, 1 = aktiviert, 0 = deaktiviert.
g_pModDisintegration	Desintegrationsmode, es gibt nur eine Waffe, die sofort killt, kann mit g_pModElimination kombiniert werden, 1 = aktiviert, 0 = deaktiviert.
g_pModElimination	Wer gekillt wurde, ist raus, bis der letzte als Gewinner feststeht, 1 = aktiviert, 0 = deaktiviert.
g_pModSpecialties	Spezialistenmodus, jeder Spieler hat eine Klasse mit besonderen Fähigkeiten und Waffen, 1 = aktiviert, 0 = deaktiviert.

Variable	Bedeutung
g_teamforcebalance	Soll der Server teams zahlenmäßig automatisch ausgleichen? 1=ja, 0=nein
g_teamautojoin	Soll ein einsteigender Spieler automatisch einem Team zugeteilt werden? 1=ja, 0=nein
g_friendlyfire	Ist friendly fire an? 1=ja, 0=nein
g_warmup dauer	Dauer Warmup, das nicht gewertet wird, Angabe dauer in Sekunden, default=5
g_dowarmup	Soll das Warmup eingeschalten sein 1=ja, 0=nein
g_quadfactor	Setzt die Verstärkung für „quad damage“ auf den angegebenen Faktor
g_filterban	Bans aktivieren
bot_minplayers	Mindestanzahl Spieler *pro Team*, auf die ggf. mit Bots aufzufüllen ist
bot_nochat	Sollen Bots *keine* Chatmessages raus hauen? 1=ja, 0=nein
g_holointro	Soll beim Beitreten das Holodeck-Intro angezeigt werden? 1=ja, 0=nein
g_weaponrespawn	Zeit in Sekunden, bis aufgesammelte Waffen wieder erscheinen
sv_allowdownload	Sind Downloads vom Server erlaubt? 0=nein, 1=ja, Standard, 2=?, 4=UDP-Downloads sind deaktiviert, 8=Keine Verbindungsunterbrechung des Clients während dem Download
cl_allowdownload	Sind Downloads durch den Client erlaubt? 1=ja, 0=nein
sv_dlURL	Web-/FTP-Adresse, aus der Maps nachgeladen werden sollen, statt vom Server direkt
g_knockback	Rückstoß
cg_thirdperson 1/0	Third-Person-Modus aktivieren/deaktivieren
cg_thirdpersonrange x	Abstand zur Spielfigur einstellen
cg_drawtimer 1	Timer für die Restzeit anzeigen lassen (ohne Limit wird die bisher gespielte Zeit angezeigt)

Unklare Variablen:

Variable	Bedeutung
teamflags	team game flag
dmflags	deathmatch flags
sv_running	server status

- weitere Quake 3 Befehle
- Weitere (neue) cvars in Verbindung mit ioQuake-abgeleiteten EF-Versionen

Netzwerk/Internetserver

Netzwerkeinstellungen/Netsettings

Im Menüpunkt Sound/Netzwerk gibt es eine Kleinigkeit, die gerade beim Multiplayer ihre Reize zeigt: Datenrate sollte auf LAN/KABEL/xDSL gestellt sein, damit das Spiel mit maximaler Netzwerkleistung gespielt werden kann. So nebenbei kann es auch den Map-Download beschleunigen, aber um Maps sollte man sich mit original EF vor dem Spielen kümmern (das dauert in EF einfach zu lange).

Wer es genauer haben will, kann auch direkt an den Netsettings drehen. Die Defaultwerte sind für

schwachbrüstige Verbindungen wie 56k Modem schon OK, aber in Zeiten von Breitbandanschlüssen kann man sich oft eine Menge Laggs sparen, indem man die Werte selbst festlegt. Für alle Werte gilt: Es hängt im Zweifelsfalle von der eigenen Internetanbindung ab, welche Werte sinnvoll sind. Wenn man es z. B. mit der `rate` oder den `snaps` übertreibt, kann es sein, dass man sich gerade deshalb Laggs einhandelt, weil die eigene Verbindung ausgelastet ist. Hier trotzdem mal ein paar Richtwerte, die bei nicht all zu knapper Bandbreite gute Ergebnisse liefern sollten:

- `snaps "40"`: wie oft pro Sekunde wird die Welt vom Server zum Client aktualisiert. Es macht wenig Sinn, hier mehr als die erreichbare FPS-Zahl einzugeben. Die Obergrenze ist aber ohnehin 125. Je öfter die Welt aktualisiert wird, desto ruhiger bzw. lagärmer ist das Spiel.
- `rate "25000"`: Bandbreite für das Spiel in Bytes/sek.. Das hat auch einen Einfluss auf die Geschwindigkeit, mit der Maps heruntergeladen werden können. Je mehr, desto besser. Allerdings hängt es auch vom Server ab, wie viel er hergibt.
- `cl_maxpackets "60"`: Wie oft pro Sekunde soll der Client seinen Status auffrischen? Auch hier gilt: mehr als 125 macht keinen Sinn und mehr als die FPS-Zahl macht auch wenig Sinn. Gerade hier kann man es aber vortrefflich übertreiben, wenn der Upload nicht ausreicht, um diese Aktualisierungsrate zu leisten.
- `cl_packetdup "1"`: Wie oft sollen Pakete Wiederholt werden? Ist besonderes sinnvoll, wenn man eine Verbindung hat, die verlustreich ist (viel loss). Auch hier gilt: Man sollte es nicht übertreiben, wenn man sich nicht sicher ist, dass die Leitung das her gibt. Insbesondere in Zusammenhang mit den anderen Einstellungen kann das sehr schnell die Leitung überlasten, da mit jedem Schritt Wiederholung mehr eben auch die anderen Einstellungen ggf. mit verdoppelt werden. Über die `rate` kommt man aber nie hinaus.

(Quelle) f

Server beitreten

Die Mobile Infanterie bietet momentan drei Server für EF:

- www.mobile-infanterie.de:27960: „EF Classic @ Mobile Infanterie“, Standard, DM/CTF gemischt,
- www.mobile-infanterie.de:27961: „EF Pinball @ Mobile Infanterie“, Pinball-Mod und
- www.mobile-infanterie.de:27962: „EF Teamelite @ Mobile Infanterie“, Team-Elite Mod.

Servern beitreten ist unkompliziert. Der Menü-Weg ist folgender:

- `Multi-Match` → `Server suche`, oder
- `Multi-Match` → `Server angeben`.

Letzteres ist dann zu empfehlen, wenn man die IP und den Port des Rechners kennt, auf dem ein EF-Server läuft und vor allem dann, wenn der Server nicht unter den Standard-Ports 27960 bis 27963 läuft, oder nicht im LAN. Dann schlägt ein Server-Query im LAN nämlich fehl.

Im anderen Fall gibt es ein paar Feinheiten. Nachdem man EF nicht nur im LAN, sondern auch im Internet spielen kann, unterscheidet EF zwischen `Internet1`, `Internet2`, ..., `Lokal` und `Favoriten`. `Internet1` bis `Internetx` läuft über Masterserver, die zum Glück noch teilweise laufen. Man kann statt der ingame-Suche das Programm Pathfinder beim Server suchen für EF verwenden.

Aber für reine LAN-Server LAN muss man `Lokal` auswählen. Dabei fragt EF einfach alle Rechner im

eigenen Subnetz an, ob Server auf den Rechnern im Netzwerk laufen. (Funktionsweise ist im Abschnitt Lokale Server Abfragen via Broadcast beschrieben.) Solltet man trotz laufendem Server keinen aufgelistet bekommen, sollte der Rechner mit dem Server auf seine Firewall-Einstellungen überprüft werden (falls es eine gibt). Die Firewall muss EF natürlich generell ins Netzwerk lassen. Wenn der Server-Port im oben genannten Bereich liegt, wird EF ihn im lokalen Netzwerk finden. Anderenfalls ist es ein Netzwerkproblem und hat sicher nichts mit dem Spiel zu tun. Die Favoriten sollten selbsterklärend sein.

Man kann noch Filter aktivieren/deaktivieren, die die Suche erleichtern:

- Spieltyp (welche Art Spiel soll auf dem Server gespielt werden)
- Volle Anzeigen (wenn ja, werden auch Server angezeigt, auf denen kein Platz mehr frei ist)
- Freie Anzeigen (wenn ja, werden auch Server angezeigt, auf denen gerade kein Spieler ist)

Sortieren kann man die Liste auch, aber im LAN wird es da nicht so viele geben.

Falls man auf eine Anfrage mal keine Antworten erhalten hat (keine Server, obwohl einer läuft), versucht man es einfach nochmal mit dem Menüpunkt `Update`. Kleines Detail: Sollte die Serversuche sofort wieder beendet sein, man erhält aber keine Spieleserver, war zumindest die Abfrage selbst wohl zugestellt worden, also der Masterserver hat etwas geantwortet. Andernfalls (Masterserver down) dauert es einen Moment, bis EF den Timeout abgewartet hat. Dieser Umstand kann helfen, das eigentliche Problem einzukreisen. Solltet man einen Spieleserver im Internet (oder auch im LAN) besonders gut finden, kann man ihn mittels der Menüfunktion `Favoriten hinzfüg.` in die Favoriten aufnehmen. Das spart u. U. Zeit, weil Server-Abfragen und dergleichen weg fallen können. Wenn man auf einen laufenden Server klickt, wird links oben die aktuell gespielte Karte angezeigt.

Server aufstellen und nutzen

Über das GUI

Weiter oben steht eine Menge über Maps und deren Auswahl. Hier steht nun noch etwas mehr dazu, wie man einen Spieleserver erstellt. Nachdem das Solomatch (Singleplayer) nur ein Sonderfall vom Multi-Match (Multiplayer) ist, gibt es dazu keine gesonderten Erklärungen. Für einen LAN-Server gibt es grundsätzlich zwei Alternativen: Entweder man macht einen sogenannten `Dedizierten Server`, oder auf Englisch `Dedicated Server` auf, oder man macht einen sogenannten `Listen-Server` auf. Dazu gibt es wieder zwei Methoden: Entweder man nimmt ein extra Programm dafür her (das zumeist für Dedicateds bessere Konfigurationsmöglichkeiten hat), oder man nimmt EF selbst dafür. Nachdem es viele Programme dafür gibt, erkläre ich nur wie man das mit EF selbst macht. Das hat schließlich jeder. Falls aber doch noch einer von euch Lust hat ein Programm dafür zu benutzen, kann ich den EF-Runner empfehlen. Den erkläre ich hier aber nicht zusätzlich. Der erstellt dann die notwendigen Konfig-Files, ggf. auch für mehrere Server.

1. Der Menü-Weg ist folgender: `Multi-Match` → `Server erstellen`
2. Spieltyp wählen:
    - `Frei für alle` (FFA = DM)
    - `Team-Holomatch` (TDM)
    - `Tournament` (1 on 1)
    - `Capture the Flag` (CTF)
3. Map auswählen (das ist die nächste, also die, welche direkt beim Einstig gespielt wird)
4. ggf. Modifikationen auswählen (Elimination, Spezialist, ActionHero, Disintegration, Assimilation).

Nicht alle Kombinationen an Modus und Modifikation sind möglich. Z. B. verlangt Assimilation zwingend Teamdeathmatch als Modus.

5. Auf `weiter` klicken
6. Hostname eingeben: Den sehen die anderen nachher bei der Server-Suche in der Liste
7. Bei `Spieler` unter `Mensch` evtl. noch die persönlichen Daten ändern wenn man das noch nicht getan hat. (Das gibt es natürlich nicht, wenn man den Server `dediziert` startet.)
8. Auswählen, wie viele offene, bzw. mit Bots belegte Slots zur Verfügung stehen sollen.
9. `HC-Stufe`: Schwierigkeitsgrad der Bots
10. Punkte und Zeitlimit festlegen
11. Evtl. erweiterte Einstellungen setzen
12. Server festlegen: Bei der Option `Nur Server` handelt es sich *nicht* um die Dedicated-Funktion, sondern um eine Sicherheits-Funktion, die folgendes bewirkt: Ist sie an, prüft der Server, ob der Client (= Mitspieler) auch wirklich exakt die selben Dateien hat, wie auf dem Server (`sv_pure = 1`). Das heißt, u. A. dass man nicht mit veränderten `*.pk3`-Dateien mogeln kann. Braucht man aber normalerweise auf LANs nicht. Da sieht man ja, ob jemand cheatet. Allerdings ist das mitunter der Grund, weshalb man an den Original `*.pk3`-Dateien nichts verändern sollte, auch nicht neu komprimieren, oder so. Die Datei ist danach evtl. nicht mehr brauchbar… Unter `Dedizierter Server` hat man drei Einstellungsmöglichkeiten:
    - `NEIN` (= Kein dedicated Server, also = Listen-Server)
    - `LAN` (nur Leute aus dem lokalen Netzwerk können einsteigen)
    - `INTERNET` (Leute aus dem Internet können ebenfalls einsteigen, nicht zu empfehlen auf reinen LANs)

Zur Benutzung eines Dedicated Server auf dem eigenen Rechner und parallel Spielen müsste man eine weitere Instanz von Elite Force starten, um sich in den Server einzuloggen. Nachdem das aber einen schnellen Rechner braucht, sodass das keine Probleme verursacht, empfiehlt es sich hierfür einen Rechner abzustellen, der das Ding laufen hat, oder lieber einen Listen-Server zu starten. Und noch eine Einschränkung hat der in EF erstellte Dedicated-Server: Man kann immer nur eine Map durchlaufen lassen, außer man kennt sich ein wenig mit Skripten aus (aber das führt hier zu weit). Programme wie der EF-Runner können auch eine Mapcycle ohne Probleme erstellen und das ausgesprochen komfortabel.

Via Config-File

Es gibt grundsätzlich zwei Möglichkeiten, wie man Server automatisiert starten lassen kann, ohne sich vorher durch Menüs klicken zu müssen. Die eine läuft über ein Start-Skript ab, also unter Windows eine `*.bat`-Datei, unter Linux ein Shell-Skript. Beide nutzen dann aber eine Server-Config, in der letztlich steht, wie der Server auszusehen hat und was er zu tun hat. Eine minimale Server-Config könnte z. B. so aussehen:

```
seta hostname "Mein Server"
seta sv_maxclients "32"
seta fraglimit "30"
seta timelimit "15"
seta capturelimit "5"
seta g_friendlyfire "1"
map hm_voy1
echo "Laeuft!"
status
```

Eine etwas erweiterte Config könnte so aussehen (anklicken zum Ausklappen): <spoiler>

```
seta sv_hostname  "your server name"

seta g_motd "your message of the day"

sets Aministrator "your name here"
sets clan "your clan here"
sets ClanWebsite "your web site here"
sets Email "your Email here"
sets location "where you are"
sets CPU "your PC discription here"
sets OP "your operating system here"
sets RAM "your Amount of ram here"
sets Connection "your connection here"

//Double slashes are used for comments and to keep commands from running.
Remove the slashes before a specific to run that command.
//Default Capture The Flag Server Configuration File

//These settings need to be set as they are. You can lower the maxrate, but
8000 is our ceiling
set bot_enable 0
set sv_floodprotect 1
Set g_gametype 4
set sv_maxRate 8000
set sv_maxclients 12

//set to 1 to enable expansion mod type
set g_pMODElimination 0
set g_pMODActionHero 0
set g_pMODDisintegration 0
set g_pMODAssimiliation 0
set g_pMODSpecialties 0

//These are recommended settings
set g_holoIntro 0
set sv_allowdownload 1
set g_doWarmup 0
set sv_pure 1
set capturelimit 10
set fraglimit 15
set timelimit 30
set g_friendlyFire 0
set sv_privateClients 0
set sv_privatePassword ""
set g_syncronousClients 0

//the EF master server is already set
set sv_master1 "efmaster.tjps.eu"
set sv_master2 "master.stvef.org"
```

```
set sv_master3 "master.stef1.ravensoft.com:27953"
seta public 1

//Set map selection, cycle order, and start first map

set d1 "map hm_borg1 ; set nextmap vstr d2
set d2 "map hm_borg2 ; set nextmap vstr d3"
set d3 "map hm_borg3 ; set nextmap vstr d4"
set d4 "map hm_kln1 ; set nextmap vstr d5"
set d5 "map hm_scav1 ; set nextmap vstr d6"
set d6 "map hm_dn1 ; set nextmap vstr d7"
set d7 "map hm_dn2 ; set nextmap vstr d8"
set d8 "map hm_for1 ; set nextmap vstr d9"
set d9 "map hm_noon ; set nextmap vstr d10"
set d10 "map hm_cam ; set nextmap vstr d11"
set d11 "map hm_voy1 ; set nextmap vstr d12"
set d12 "map hm_voy2 ; set nextmap vstr d13"
set d13 "map ctf_kln1 ; set nextmap vstr d14"
set d14 "map ctf_kln2 ; set nextmap vstr d15"
set d15 "map ctf_and1 ; set nextmap vstr d16"
set d16 "map ctf_voy1 ; set nextmap vstr d17"
set d17 "map ctf_voy2 ; set nextmap vstr d1"

vstr d1


//Misc Settings
set g_weaponrespawn 0
set g_adaptrespawn 1
set g_allowvote 1
set _quadfactor 3

set g_intermissionTime 20
//set g_ghostRespawn 3
//set g_team_group_red hazardteam
//set g_team_group_blue borg
seta g_password ""
seta rconpassword ""
seta sv_maxclients 12
seta sv_minclients 0
seta g_maxgameclients 12
seta sv_privateClients 0
seta sv_privatePassword ""
seta sv_maxRate 8000
seta sv_maxping 0
seta sv_minping 0
seta g_intermissionTime 20
seta g_ghostRespawn 1
seta g_forcespawn 5
seta g_inactivity
seta g_doWarmup 20
```

```
seta g_warmup 20
seta g_holointro 1
seta sv_pure 1
seta g_allowvote 0
seta sv_allowdownload "1"
seta g_friendlyfire 0
seta g_TeamForceBalance 1
seta g_TeamAutoJoin 0
seta sv_floodProtect 1
```

</spoiler> Für eine Mapcycle muss, wie im zweiten Beispiel gezeigt, manuell festgelegt werden, welche Map nach der jeweiligen aktuellen geladen werden soll und am Ende wieder die erste und diese Cycle auch ein mal gestartet werden. Man kann hier auch zusätzlich, neben der nächsten Map auch den Modus für die aktuelle Map und andere Einstellungen setzen, die sich beim Mapwechsel ändern sollten. Auf diese Weise kann man z. B. zwischen CTF und DM hin und her wechseln, je nach Map.

Im Prinzip ist der Name für die Config egal. Als `autoexec.cfg` wird die Config aber automatisch geladen, wenn man nichts anderes angibt. Die manuelle Angabe kann dann so aussehen: `„C:\Program Files\Raven\Star Trek Voyager Elite Force\stvoyHM.exe“ +seta dedicated 2 +seta sv_pure 0 +exec autoexec.cfg`. Das kann man so entweder als Start-Befehl in eine Windows-Verknüpfung packen (der Pfad, in welchem das Programm ausgeführt werden soll, muss entsprechend auch angepasst sein), oder in das Start-Skript. Für Linux sieht die Kiste entsprechend auch aus: `./binaryname +seta dedicated 2 +seta sv_pure 0 +exec autoexec.cfg` (vom Verzeichnis ausgehend, in dem der EF Server liegt). Mit einer EF-Installation unter Windows ist hier nicht mehr viel zusätzlich zu tun. Wenn man aber einen nackten Server aufsetzen will, braucht man nicht alles davon. Man braucht nur von der CD das Verzeichnis `baseEF`, das man unter Linux aber in `baseEF` umbenennen muss, da der Server sonst seine Dateien nicht findet. Dazu noch die `stvoyHM.exe`, bzw. eine entsprechende Dedicated Binary, die *nicht* in `baseEF` liegt, sondern eine Ebene höher. Config-Dateien muss man in `baseEF` ablegen, auch wenn man den Dedicated mit Mods aktiv starten will (dort sucht der Server allgemein seine Dateien, u. A. auch die Maps, bevor es los geht). Mancher Dedicated Server sucht unter Linux allerdings im Ordner `~/.stvef/baseEF` danach. Möchte man einen Dedicated nicht mit Standard EF sondern mit einem Mod aktiv starten, muss man dies beim Start mit übergeben, z. B. für Pinball durch `+set fs_game pinball +set dedicated 2 +exec pinconfig.cfg`. Außer `exec` sollten die anderen Parameter auch in einer entsprechend gestarteten Konfig gesetzt werden können. Für die Benutzung von Mods ist die Angabe `fs_game` dringend *als Startparameter* notwendig (also nicht in der Konfig setzbar). Der Name des Mods entspricht normalerweise dem Verzeichnis, in dem die Mod-Daten liegen. Man könnte gewissermaßen sagen, für Standard EF ist der Mod `baseEF` (und muss nicht extra angegeben werden), weil dafür in diesem Verzeichnis die Daten gesucht werden. Unter Windows muss man dann nicht mehr viel machen, und das Skript ggf. mit dem richtigen Config-Namen versorgen. Unter Linux muss man im Prinzip auch nicht viel machen, nur ggf. die Dedicated-Binary ausführbar machen. Für weitere Einstellungen in der Config, siehe die Config-Parameter.

Hier eine kurze Übersicht, welche Modifikationen und welche Modi zu welchen Config-Variablen gehören:

Modus

Modusname	g_gametype
Free for all (DM)	0
Tournament	1
Solo*	2
Team Deathmatch	3
Capture the Flag	4

Benutzt man andere Zahlen, wird der Spielmodus auf FFA gesetzt.

* Anmerkung: Solo steht nur der Vollständigkeit halber hier und ist im eigentlichen Sinne kein Modus für Server. Dieser Modus wird gesetzt, wenn man in Elite Force via GUI ein Solomatch startet, also gerade ohne menschliche Spieler. Dieser Modus ist im Prinzip auch via Konsolenparametern bzw. in einer Config benutzbar, aber es ist dann eben nicht möglich, dass andere Spieler einsteigen können. Er ist entsprechend auch nicht via Netzwerk erreichbar (auch nicht über `localhost` / `127.0.0.1`).

Modifikation	Variable
Assimilation	`g_pModAssimilation = "1"`
Elimination	`g_pModElimination = "1"`
Specialties	`g_pModSpecialties = "1"`
ActionHero	`g_pModActionHero = "1"`
Disintegration	`g_pDisintegration = "1"`

Erlaubte Kombinationen an Modifikationen und Modi

Es sind nicht alle Kombinationen aus Modifikation und Modus erlaubt. Nachfolgend steht eine Übersicht, welche Kobinationen der Modi mit Modifikationen und der Modifikationen untereinander erlaubt sind.

Modus und Modifikation

Modus	Assimilation	Specialties	Disintegration	Elimination	Actionhero
Free for all	nein	ja	ja	ja	ja
Team Deathmatch	ja	ja	ja	ja	nein
Tournament	nein	ja	ja	ja	ja
Capture the flag	nein	ja	ja	nein	nein

Modifikationen untereinander

	Assimilation	Specialties	Disintegration	Elimination	Actionhero
Assimilation	ja	ja	nein	nein	nein
Specialties	ja	ja	nein	ja	nein
Disintegration	nein	nein	ja	ja	nein
Elimination	nein	ja	ja	ja	nein
Actionhero	nein	nein	nein	nein	ja

Erreichbarkeit

Für EF muss man, falls man selber einen Server erstellt, in einer Firewall lediglich den Port 27960 (UDP) freigeben (oder alternativ den selbst gewählten). Zum Beitreten sollte man eigentlich keinen bestimmten freigeben müssen. Es gilt jedoch beim Erstellen des Server zu unterschieden zwischen LAN-Server und Internet-Server. LAN-Server sind, wie der Name sagt, nur via LAN erreichbar, nicht über das Internet. Man kann im Prinzip auch mehrere Server laufen lassen, indem man mehrere Ports wählt. Einige Ports können automatisch von EF abgesucht werden, sodass z. B. auch ein parallel laufender zweiter Server nicht manuell in EF eingegeben werden muss. Die automatisch gescannten UDP-Ports für LAN-Suche sind 27960-27963. Diese müssen bei Benutzung logischerweise auch freigegeben werden, wenn man hinter einer Firewall oder einem NAT sitzt.

Masterserver

Der Masterserver von Eliteforce wird automatisch über neu erreichbare Server benachrichtigt, sobald sie starten. Das allerdings nur, wenn man den Server-Modus auf Internet stellt, z. B. über `+set dedicated 2` beim Start auf der Konsole. Mit `+set dedicated 1` wird zwar der Server gestartet, aber nicht bekanntgegeben. Clients wiederum laden sich vom Masterserver die ihm bekannten Spieleserveradressen runter. Die zu verwendenden Masterserver werden in den Configfiles von Client (`hmconfig.cfg`) und Server (je nach dem, wie man sie eben benannt hat) über `seta sv_master1 "blahfasel"` bis `sv_master5…` gesetzt. Die Liste der aktuell zu findenden Server ist hier auch zu sehen.

aktuelle Masterserver

Hier ist eine Liste der momentan laufenden Masterserver der Community:

- `master.stvef.org` (2019-06-09), IP 74.91.116.133
- `efmaster.tjps.eu` (2019-06-09), IP 176.9.53.212
- `master.stef1.daggolin.de` (2019-06-09), IP 185.7.199.7

Diese sollten sowohl in den Server- als auch in den Client-Configs eingetragen werden. Um einen eigenen Masterserver aufzustellen, könnte dieses Projekt hilfreich sein. Unter Debian Stretch x64 konnte ich den Code allerdings nicht compilieren, offenbar wegen eines Problems, das schon 2007 bekannt war. Der Standard-Port dafür ist 27953 UDP. Alternativ könnte auch dieses Projekt für EF nutzbar sein. Um zu prüfen, ob ein Masterserver überhaupt antwortet/erreichbar ist, kann es hilfreich sein, dieses Tool zu verwenden. Der Befehl lautet dann `qstat -efm <servernameoderip>`.

ehemalige Masterserver

- `master.stef1.ravensoft.com` (2017-02-03), IP 104.40.23.123
- `master.gamespy.com:27900`
- `efmaster.kickchat.com` (2017-03-03), IP 176.9.53.212

Der GameSpy-Server ist nach dem Ableben des Netzwerks logischerweise nicht mehr existent. Funktioniert haben dürfte er aber noch nie, wegen des im Abschnitt Masterserver Abfrage Bug beschriebenen Problems.

Masterserver Abfrage Bug

Im original Eliteforce Spiel bis einschließlich Version 1.2 gibt es einen Bug, der es zwar erlaubt, einen Dedicated-Server mit Angabe einer Portnummer für den Masterserver bekannt zu machen. Z. B. kann über den per Default vorhandenen Masterserver `master.gamespy.com:27900`, ein Spieleserver bekannt gemacht werden. Allerdings können die Clients mit Version 1.2 oder älter den entsprechenden Masterserver überhaupt nicht erreichen, weil sie es *immer* auf Port 27953 versuchen, egal was im Konfig-File steht. Deshalb ist es auch ratsam, Masterserver, die (auch) EF-Server listen sollen, *nur* auf Port 27953 zu konfigurieren. Neuere Versionen, die auf ioQuake3 basieren, sind von diesem Problem nicht betroffen und können auch die 1.2er Server erreichen, die sich über einen anderen Port bekannt gemacht haben.

Technisches Konzept der Masterserver

Im Prinzip ist das Konzept der Masterserver ganz einfach: Ein dedizierter Server nimmt Anfragen entgegen, welche Spieleserver er denn kennt. Diese Liste gibt er dann dem anfragenden Client zurück. Das ist das was passiert, wenn man in der Serversuche die Einträge „Internet 1“ bis „Internet 5“ (oder mehr?) nutzt.

Anders herum muss der Masterserver überhaupt erst mal Kenntnis von der Existenz eines Spieleservers erlangen (Listen- und LAN-Server werden nicht aufgeführt, nur Internet-Server, was in den Einstellungen eines Servers durch die Variablendefinition `dedicated = 2` konfiguriert wird). Dazu meldet sich ein entsprechend eingestellter Spieleserver beim Masterserver durch einen sogenannten „Heartbeat“. Dieser enthält letztlich nur die Information „Huhu, hier bin ich, nimm mich bitte in deine Liste auf!“ Der Master nimmt aber nur Server auf, die auch erreichbar sind. Das prüft er, indem er einmal nach so einer Meldung den Status des gerade sich meldenden Server abfragt (liefert dann u. A. Spieleranzahl, Version, etc…). Erhält er dabei keine Antwort, nimmt er den Server nicht auf. Das muss man z. B. bedenken, wenn man einen Server hinter einer Firewall oder einem NAT betreibt. Ist der Server-Port zu, erreicht der Masterserver ihn nicht, damit taucht er auch nicht in der Liste auf.

Wenn man es genau nimmt, werden nur die Server zurück gemeldet, die der Version des Anfragenden entsprechen, ggf. eingeschränkt auf keine vollen oder keine leeren Server. Letztlich ist das Prinzip sehr einfach gehalten und hat auch einige kleinere Anfälligkeiten. So basiert das Protokoll (genau wie das Spiel-Protokoll) auf UDP. Heißt, wenn eine Anfrage oder Antwort verloren geht, merkt das keiner. Ggf. ist also alles an beiden Enden völlig OK, aber man kriegt trotzdem als Client keine Server gelistet. Wie genau das technisch funktioniert, ist im folgenden Beschrieben.

Protokoll dahinter

Das Protokoll ist weitestgehend mit dem von Quake III Arena identisch. Einige kleinere Änderungen gibt es allerdings.

Im Folgenden werden Server-Kommunikations-Strings gezeigt, die alle mit vier mal 0xff starten, der Zeichenkolonne ÿÿÿÿ. Sollte beim Leser diese Zeichenkolonne falsch dargestellt werden (sieht eigentlich aus wie vier y mit Punkten oben drauf), bitte nicht verwirren lassen. Es sind vier mal das ASCII-Zeichen Nr. 255 (also alle 8 Bits auf 1 gesetzt) hintereinander, keine Lücken.

Spieleserver beim Masterserver melden (Heartbeat)

Ein Spieleserver kann sich beim Masterserver durch einfaches Zusenden der Zeichenfolge `ÿÿÿÿ\heartbeat\27960\gamename\STEF1\` melden. Die Ziffernkolonne `27960` entspricht dem Port des sich meldenden Servers und kann entsprechend abweichen. Der Masterserver wird daraufhin versuchen, den Serverstatus *des Absenders* abzufragen. Gelingt das, wird der Server aufgenommen. Herangezogen werden als gespeicherte Informationen die IP, der Port und das Protokoll des Spieleservers, sowie die Information ob er leer oder voll ist. Das hat einmal den Sinn, dass man dem Client nur Server meldet, auf denen er spielen kann. Für Eliteforce 1.2 oder neuer ist das Protokoll Nr. 24 (bzw. für ältere Versionen wie 0.28 kann es auch 22 sein). Zum anderen gibt der Client bei der entsprechenden Abfrage des Masterservers bereits an, ob er volle und/oder leere Server gemeldet bekommen will. Der Anfrage entsprechend gefiltert wird dann auch geantwortet.

Danach finden erst mal keine weiteren Dinge statt. Es findet keine Antwort an den sich meldenden Spieleserver statt. Der Master prüft u. A. auch nicht anlasslos, ob der Spieleserver noch läuft. Für diesen Zweck gibt es einen einfachen Ersatzmechanismus: Wenn ein Server beendet wird (Kommando `quit`) sendet er nochmal einen letzten Heartbeat und geht dann offline. Da nun der Masterserver bei der Abfrage des Serverstatus keine Antwort bekommt, geht er davon aus, dass der Server tot ist, und gibt ihn bei Client-Anfragen nicht mehr aus.

Der Spieleserver hingegen meldet sich in regelmäßigen Abständen beim Masterserver. Das hat auch den Effekt, dass eine ggf. nicht vom Masterserver empfangene Meldung nach einiger Zeit (hoffentlich) doch irgendwann erhalten wird, und der Spieleserver daraufhin gelistet wird.

Ebenfalls meldet sich der Spieleserver für die Übergänge Leer → Nicht Leer und Voll → Nicht voll, bzw. umgekehrt. Dazu kann man auf der Konsole (z. B. via rcon) manuell einen Heartbeat senden. Es gibt also alles in allem acht Situationen, in denen ein Heartbeat gesendet wird:

1. Server geht das erste mal online,
2. Server geht offline,
3. Serverzustand geht von leer nach besetzt über,
4. Serverzustand geht von besetzt zu leer über,
5. Serverzustand geht von nur besetzt zu voll über,
6. Serverzustand geht von voll zu nur besetzt über,
7. Konsolen-Befehl `heartbeat` wird abgesetzt und
8. Server gibt routinemäßig alle paar Minuten ein Lebenszeichen.

Spieleserver beim Masterabmelden (Heartstop)

Es gibt in Eliteforce offenbar einen zusätzlichen Mechanismus, der wohl dafür gedacht ist, sich beim Masterserver explizit abzumelden. Dieser wird mit dem folgenden String veranlasst: `ÿÿÿÿ\heartstop\27960\gamename\STEF1\`. Auch hier ist die Zeichenkolonne `27960` der Port des sich abmeldenden Spieleservers. Diese Abmeldung ist aber im eigentlichen Sinne nicht nötig und weicht von der originalen Quake 3 Implementation ab, siehe vorangegangener Abschnitt Spieleserver beim Masterserver melden (Heartbeat).

Serverlistenabfrage

Die Abfrage der dem Masterserver bekannten Spieleserver durch einen Client wird so eingeleitet: `ÿÿÿÿgetservers xx`, mit `xx` als einer numerischen Angabe der Spieleprotokollversion. Für Eliteforce ist das seit Version 1.2 24. Die Version 0.28 hatte noch als Angabe 22. Weiterhin kann die Abfrage um die (mit Leerzeichen 0x20, ASCII-Zeichen Nr. 32 abgetrennten) Erweiterungen `full` (gibt auch volle Server zurück) und `empty` (gibt auch leere Server zurück) ergänzt sein. Eine Abfrage, die alle Eliteforce 1.2 Server zurück gibt, die der Masterserver kennt, sähe also so aus: `ÿÿÿÿgetservers 24 empty full`.

Serverlistenantwort

Die Antwort an den Client sieht dann so aus: `ÿÿÿÿgetserversResponse \adresseport1\addresseport2\…\adresseportn\EOT`. Der Block `ÿÿÿÿgetserversResponse` und der Block `\EOT` sind Pflichtprogramm. Dazwischen stehen ggf. (sofern es bekannte Server gibt) die Adressen und Ports der bekannten Server. Jeder Serverblock hat die Form `\ipPort`. `ip` meint hier acht Bytes der IP in Kleinbuchstaben-Hex-Darstellung. Der erste Adressblock entspricht dem ersten Hex-Paar. Die Adresse `192.168.0.1` wird so zu `c0a90001`. Der Block `Port` wird direkt danach angegeben. Das ist der Serverport als zwei Bytes little endian unsigned short, oder kurz, so wie man die Hex-Zahl auch auf den Zettel schreiben würde. Aus dem Standard-Port 27960 wird dann z. B. `6d38`. Beide Werte (IP-Blöcke und Port) werden ggf. mit führenden Nullen dargestellt, um die Anzahl der Zeichen bei acht bzw. vier zu halten. Der gesamte Server würde also wie folgt ausgegeben: `\c0a900016d38`. Die Gesamte Antwort für die Server `192.168.0.1:27960`, `192.168.178.1:27961` und `123.34.56.78:9012` sähe dann so aus: `ÿÿÿÿgetserversResponse \c0a900016d38\c0a9b2016d39\7b22384e2334\EOT`. Zu beachten ist hier nochmal, dass nur die Server angezeigt werden, die dieselbe Version haben, wie der Client (bei EF 1.2 und neuer also 24) und die leeren bzw. vollen nur dann, wenn der Client dies ausdrücklich anfordert. Andere Server werden nicht zurück gemeldet.

Abfrage eines Spielserver Status

Wie oben bereits beschrieben, fragt der Masterserver den sich meldenden Spieleserver vor dem Hinzufügen wenigstens einmal ab, statt ihn einfach ungetestet hinzuzufügen. Die entsprechende Abfrage an den Spieleserver sieht so aus: `ÿÿÿÿgetinfo xxx`. Wobei `xxx` so wie es da steht, bereits funktioniert. Für eine echte Statusabfrage kann ggf. auch noch eine Challenge-Nummer (Ziffernfolge) dort stehen. Für den Masterserver ist diese aber belanglos.

Antwort Spielserver Statusabfrage

Der Spieleserver antwortet dann mit allerlei Informationen über sich. Von der Grundstruktur her sieht das dann so aus:

```
ÿÿÿÿinfoResponse "Werte"
```

Wobei die Werte zwischen den "-Zeichen zumeist mehr Informationen enthalten, als für den Masterserver notwendig sind. Deren Struktur ist stets `\Bezeichner\Wert`. Als letztes Bezeichner-Wert-Paar steht dort stets `\challenge\xxx`, die bereits erwähnte Challenge. Bei Masterserverabfragen ist der Wert `xxx` wörtlich zu verstehen, bei echten Status-Abfragen von Clients kann das anders aussehen. Bereits angetroffenen Werte sehen wie folgt aus:

- game: Welcher Mod wird gespielt, also z. B. standard baseEF, oder aber auch pinball.
- voip: Welcher Voice over IP Codec wird verwendet, z. B. opus.
- g_needpass: Verlangt der Server ein Passwort zum Zugang? 1 = ja, 0 = nein.
- pure: Wert der Variable sv_pure auf dem Server. Falls 1, sind dem Client nur dieselben *.pk3-Dateien erlaubt, die auch der Server vorliegen hat.
- gametype: Welcher Gametype ist gesetzt, z. B. 4 für CTF. Entspricht der Server-Variablen g_gametype. Siehe auch Abschnitt Modus.
- sv_maxclients: Wie viele Clients dürfen maximal verbunden sein? Ziffernfolge. Wird für die Ermittlung des Zustands full (voll) bei der Abfrage am Masterserver herangezogen.
- g_humanplayers: Wie viele tatsächlich menschliche Spieler spielen auf dem Server? Ziffernfolge.
- clients: Wie viele Spieler sind gerade auf dem Server unterwegs? Enthält auch die Bots, im Gegensatz zu g_humanplayers. Wird für die Bestimmung des Zustands full (voll) und empty (leer) herangezogen.
- mapname: Der technische(!) Name der Map, die gerade gespielt wird, z. B. hm_voy1.
- hostname: Welche Bezeichnung wurde dem Server gegeben? Z. B. EF Classic @ Mobile Infanterie.
- protocol: Welche Version hat der Spieleserver? Für EF ab 1.2 24, alte Versionen haben aber z. B. auch 22 (Version 0.28).
- gamename: Welches Spiel wird gespielt? Für EF ist das EliteForce.

In der Standard-Implementation ist es nicht notwendig, dass der Masterserver die Spieleserver vor Senden der Clientliste nochmal testet. Da aber eine Abmeldung nicht zwangsläufig beim Server ankommen muss (UDP), kann es ggf. sinnvoll sein, in regelmäßigen Abständen eine Abfrage an die Server zu senden und nicht mehr erreichbare Server auszusortieren.

Lokale Server Abfragen via Broadcast

Im LAN sind im Allgemeinen keine Masterserver notwendig, da die Server einfach via Broadcast-Nachrichten aufgespürt werden. Dazu wird die im Abschnitt Abfrage eines Spielserver Status beschriebene Abfrage statt an einen bestimmten Server an die Broadcast-Adresse 255.255.255.255 gesendet. Das hat zur Folge, dass das Abfragepaket bei allen Teilnehmern ankommt. Wer im richtigen Subnetz sitzt, wird sich daraufhin angesprochen fühlen und antworten, sofern ein Internet- oder LAN-Server läuft. Auch hier werden alle Server mit Ports zwischen 27960 und 27963 abgefragt.

Sollte ein Spieleserver hinter einer Firewall, einem NAT oder auch einer Desktop-Firewall sitzen, die entweder den Spieleserver-Port oder aber Broadcast-Anfragen blockt, wird der Server nicht im Server-Browser auftauchen. Server die auf anderen Ports erreichbar sind, als zwischen 27960 und 27963, werden ebenfalls nicht sichtbar sein. Deshalb empfiehlt es sich ohne Masterserver in diesem Bereich zu bleiben.

Abfrage von Server-Details

Oben genannte Abfragen sind alle im Kontext der Identifikation vorhandener Server zu sehen. Die Spieleserver sind aber durchaus in der Lage, noch weitere Informationen an den Client zu liefern, z. B. wer sich dort momentan aufhält, oder welche Map-Limits gesetzt sind. Die entsprechende Anfrage sieht so aus: ÿÿÿÿgetstatus. Die Antwort hat die folgende Form: ÿÿÿÿstatusResponse \Bezeichner1\Wert1\Bezeichner2\Wert2…\Bezeichnern\Wertn. Der Zeilenumbruch nach Response ist Zeichen 0x0a, bzw. ASCII Nr. 10. Die Wertepaare ähneln stark denen bei einer

Serverstatus-Abfrage. Die Liste verfügbarer Bezeichner enthält u. A. folgende Werte-Paare:

- `protocol`: 24, die Netzwerkversion des Servers.
- `difficulty`: Der Schwierigkeitsgrad der Bots.
- `unlagged`: Der Server hat den Modus „unlagged“ implementiert. Wenn dieser Bezeichner auftaucht, ist er normalerweise auch gleich auf 1 gesetzt.
- `sv_maxPing`: Maximaler erlaubter Ping von Spielern. Ist der Wert 0, so ist keine Grenze gesetzt.
- `sv_minPing`: Minimaler erlaubter Ping von Spielern. Ist der Wert 0, so ist keine Grenze gesetzt.
- `sv_dlRate`: Wie viele kB/sek. können für den Download vom Server höchstens erwartet werden.
- `sv_minRate`: Wie viele Byte/sek. werden mindestens für die Verbindung genutzt.
- `sv_maxRate`: Wie viele Byte/sek. werden höchstens für die Verbindung genutzt.
- `sv_hostname`: Der gesetzte Name für den Server, z. B. `EF Classic @ Mobile Infanterie`.
- `sv_dlURL`: Eine Erweiterung neuerer Versionen der Server: An welcher Adresse können ersatzweise Map- oder Mod-Downloads getätigt werden, wenn der Client eine Datei nicht besitzt. Wird von Original EF nicht genutzt.
- `sv_allowDownload`: Sind Map-Downloads erlaubt? `1` = ja, `0` = nein.
- `g_speed`: Welche Laufgeschwindigkeit ist gesetzt? Default ist `300`.
- `g_friendlyfire`: Ist Friendly Fire an? `1` = ja, `0` = nein.
- `sv_floodProtect`: Hat der Server den Flood-Protect Schutz aktiviert? `1` = ja, `0` = nein.
- `sv_maxclients`: Wie viele Spieler können maximal auf dem Server spielen.
- `fraglimit`: Bei wie viel Kills eines Spielers wird eine DM-Runde beendet.
- `timelimit`: Nach wie vielen Minuten ist eine Map spätestens zuende?
- `capturelimit`: Nach wie vielen Captures/Punkten einer Partei wird eine CTF-Map beendet.
- `dmflags`: Unklar, dürfte Modifikationen für den DM-Modus geben. Default scheint `0` zu sein.
- `g_maxGameClients`: Unklar, vom Namen her wohl ein Limit, wie viele Spieler insgesamt auf dem Server spielen dürfen. Das erledigt allerdings eigentlich schon `sv_maxclients`.
- `g_gravity`: Wie stark ist die Schwerkraft auf der Map? Default ist `800`.
- `g_weaponrespawn`: Nach wie vielen Sekunden taucht eine Waffe wieder an einer Stelle auf, nachdem sie von dort genommen wurde.
- `version`: Langfassung der Version des Servers, z. B. `Lilium Voyager HM 1.39_GIT_9fcb8a0-2018-03-06 linux-x86_64 Apr 30 2018`.
- `com_gamename`: Welches Spiel wird dort gespielt? Für EF: `EliteForce`.
- `com_protocol`: Welches Kommunikationsprotokoll wird vom Server verwendet? Z. B. 24 oder 26.
- `g_gametype`: Welcher Modus wird gespielt, z. B. 4.
- `mapname`: Welche Map wird gespielt (technischer Name), z. B. `hm_noon`.
- `sv_privateClients`: Wie viele private Clientslots sind gesetzt? Default ist `0`.
- `bot_minplayers`: Auf wie viele Spieler pro Team füllt der Server die Teams mit Bots auf? Default ist `0`.
- `betadate`: Unklar. Ein erhaltener Wert lautet `Feb 22 2004/16:52:14`.
- `gamename`: Welche Modifikation wird gespielt? Z. B. `PiNBALL 2.0.0` oder `baseEF`.
- `g_pModDisintegration`: Ist der Desintegration Modus aktiviert? `1` = ja, `0` = nein.
- `g_pModElimination`: Ist der Elimination Modus aktiviert? `1` = ja, `0` = nein.
- `g_needpass`: Verlangt der Server ein Passwort zum Beitreten? `1` = ja, `0` = nein.
- `g_allowVote`: Erlaubt der Server Votes? `1` = ja, `0` = nein.

Dazu kommen noch einige weniger technische Variablen, die auch mit gesendet werden:

- `Friendly fire`: Eine Freitext-Variable (ohne technische Funktion) (Text).
- `Administrator`: Enthält den Namen des Admins, sofern gesetzt (Text).
- `Email`: Mailadresse des Admins, sofern gesetzt (Text).
- `url`: Webseite zum Server, sofern gesetzt (Text).
- `Location`: Standort des Servers, sofern gesetzt (Text).

Am Ende der Antwort befindet sich ggf. noch, durch einen Zeilenumbruch (0x0a) abgetrennt, eine Liste an aktuell auf dem Server befindlichen Spielern, in der folgenden Form: `Zahl1 Zahl2 "Nickname" + (0x0a)`. `Zahl1` ist die Anzahl der Frags, `Zahl2` dürfte der Ping des Spieler sein, `"Nickname"` ist der vom Spieler gewählte Name. Der Zeilenumbruch am Ende gilt für jeden Spieler, einschließlich des letzten. Die Antwort endet also auch mit einem Zeilenumbruch.

Map-Downloads

Eine Sache in EF ist ein wenig ambivalent: EF kann von Haus aus von Internet- oder LAN-Servern Maps und Mods runterladen (sofern die Server dazu konfiguriert sind, das zuzulassen). Das hat auf der einen Seite den Vorteil, dass man die Maps nicht erst lange irgendwo suchen und manuell runterladen und ggf. ins richtige Verzeichnis schieben muss, sondern EF sich darum kümmert. Die Kehrseite ist allerdings, dass man auf diese Weise alles mögliche untergeschoben bekommen kann. U. A. auch Dateien, die dann auf anderen Servern wegen `sv_pure = 1` nicht erlaubt sind (womit man dort dann ausgesperrt bleibt). Wer hier also auf Nummer sicher gehen will, muss in seiner Konfig den Parameter `cl_allowDownload` auf `0` setzen. Das hat aber zur Folge, dass man auf Servern, die Maps, Models oder Mods verwenden, die man nicht besitzt, sofort wieder beim Verbinden runter fliegt. Das trifft ggf. sogar ohne entsprechende Fehlermeldung ein (siehe auch der Abschnitt Probleme mit Mapdownloads). Im Zweifelsfalle macht es dann Sinn, in den Logs nachzusehen, wo das Problem liegt.

Map-Downloads beschleunigen

Üblicherweise werden custom Maps, Mods und Models (also alles, was nicht standardmäßig bei EF dabei ist) vom Server angeboten. Ob diese downloadbar sind, oder nicht, hängt in erster Linie mal von der Server-Einstellung und der Client-Einstellung dazu ab. Auf dem Server muss generell `sv_allowDownload` auf 1 (oder höher) gesetzt sein, auf dem Client `cl_allowDownload` ebenso. Sonst lädt der Client nichts runter, bzw. kriegt nichts vom Server. Dieser ingame-Download ist aber von Haus aus erst mal recht langsam. Man kann zwar mit Variablen wie `cl_chunksperframe xx` clientseitig etwas daran drehen, aber so wirklich schnell wird es damit nie, da die maximale Rate bei 25 kB/sek. liegt (Engine-Limit). Deshalb gibt es mit den ioQuake-basierten Binaries für EF die Möglichkeit, die Map-Downloads auch auf andere HTTP-/FTP-Server umzuleiten, die dann auch entsprechende Downloadraten wie von einem Webserver gewohnt ermöglichen.

Dazu muss auf Server-Seite die `sv_dlURL` gesetzt sein (sowie natürlich `sv_allowDownload "1"`), z. B. `sv_dlURL "http://deine.webseite/verzeichnis"`. Dann wird der Download auf das entsprechende genannte Verzeichnis umgeleitet. Auf dem Webserver sollten dann natürlich die benötigten Dateien im entsprechenden Ordner auch zu finden und zugreifbar sein und die gleichen Dateien sein. Entsprechend heißt, um am Beispiel `sv_dlURL "http://deine.webseite/verzeichnis"` zu bleiben, dass z. B. die Files im Ordner `verzeichnis/baseEF` (Groß- und Kleinschreibung ist wichtig!) liegen müssen. Also so, als wenn die Dateien im lokalen Spieleverzeichnis liegen, mit ihren entsprechenden Oberordnern `baseEF` oder auch `pinball`. Sonst findet der Spieler die Dateien nicht und die Verbindung wird einfach sang- und klanglos beendet. Auch wichtig: Mit Version 1.38 RC 1 geht der Curl-Download *nur*, wenn der Server

nicht via https aufgerufen wird, wohl aber mit FTP. Die Original Version 1.2 kann wiederum gar nicht via Curl Maps runterladen, sodass nur der langsame Ingame-Download bleibt.

`sv_allowDownload` erlaubt mehrere Optionen:

- `sv_allowDownload 0`: *keine* Downloads erlauben
- `sv_allowDownload 1`: Downloads erlauben
- `sv_allowDownload 4`: UDP Downloads deaktivieren
- `sv_allowDownload 8`: Client nicht um Verbindungsabbruch bitten, wenn HTTP/FTP Downloads genutzt werden.

Falls man als Server-Betreiber Sorgen hat, dass andere auch Gebrauch vom eigenen Repo machen, kann man den Webserver dazu anweisen, nur Downloads des eigenen EF-Servers zu erlauben und EF beibringen, sich beim Web-Server entsprechend zu melden: HTTP_REFERER `"ioQ3://{SERVER_IP}:{SERVER_PORT}"` auf dem EF-Server setzen. Apache's `mod_rewrite` kann nun auf Basis des HTTP_REFERER den Zugriff exklusiv dem EF-Server erlauben. Auf diese Weise können nicht andere sich beim Upload bedienen.

Die maximale Bandbreite für UDP-Downloads lässt sich mittels `sv_dlRate` setzen. Die höchste damit erreichbare Rate liegt aber trotzdem bei 1 MByte/s je Client. Curl Downloads können also immer noch schneller sein.

Probleme mit Mapdownloads

Offenbar gibt es gewisse Schwierigkeiten, wenn man den `iostvefded-1.37.i386` Server und einen Client von Thilo oder Lilium Voyager nimmt. Fehlen Dateien auf dem Client aber Mapdownloads sind nicht erlaubt (default-Wert, wenn in der Konfig nichts anderes gesetzt ist), landet man unvermittelt im Hauptmenü. Es erscheint nicht die von Version 1.2 bekannte Fehlermeldung der fehlenden `*.bsp`-Datei. Sollte man die Map also dringend benötigen, sollte man in der Konfig den Parameter `cl_allowDownload` auf 1 setzen.

In beiden Fällen kann man zur Eingrenzung des Fehlers Eliteforce über die Konsole starten. Dort werden zusätzliche Informationen ausgegeben. In dem Wust an Meldungen findet man dann normalerweise auch die Meldung, dass eine bestimmte Map oder andere Datei (z. B. Spielermodel) nicht gefunden wurde.

sv_pure und Mapdownloads mit neueren Clients

Eine besondere Kombination kann es mit Thilo's Binaries oder Lilium Voyager verunmöglichen, Maps herunter zu laden, obwohl sowohl Client als auch Server das erlauben: Hat der Server `sv_pure = 1` gesetzt, und dem Client fehlen aktuell benötigte Dateien, aber er hat auch weitere, die der Server nicht kennt, wird nichts herunter geladen. Man fliegt, wie oben beschrieben, wieder aus dem Spiel, angeblich weil eine Datei fehlt, allerdings wird sie ja nicht herunter geladen. Das lässt sich nur lösen, indem entweder auf dem Server `sv_pure` auf `0` gesetzt wird, oder aber, indem der Client die „überschüssigen“ Dateien bei sich zumindest in einen anderen Ordner schiebt.

Ports

Neben den bereits genannten Ports 27960-27963 (UDP) sollen auch die Ports 26000, 27500 und 27910 (UDP) in Verwendung sein.

Web-Links

- Wenn sie nicht so veraltet wäre, wäre es ne geile Seite (auch mit paar Tools).
- EF Patch 1.2
- EF Engine Patch 1.38 RC1 (empfohlen)
- Liste aktueller Server laut Masterserver
- Weitere Liste mit aktuell laufenden Servern, optisch etwas ansprechender gestaltet
- Server-Such-Programm Pathfinder
- EF-Runner 1.8.3 (wenn das jemand wieder findet, bitte neu verlinken)
- EF-Maps EF-World
- EF-Mods EF-World
- EF-Tools EF-World
- EF-Sourcecode EF-World
- EF Downloads TrekCore
- EF auf TrekCore
- Totally EF
- Seite zu EF 1 & 2
- Gamefront: Einige Maps, Skins, Tools, usw.
- Seite zu den Spiel-Modi (Englisch)
- Clan-Pages:
  - der-orden-des-kahless.de, ein Clan, der den Mod RPG-X spielt
  - Skunky, aka Elite Force Players Guild
  - Starfleet Marine Corps
- Basiskonfiguration für einen Elite Force Linux Dedicated Server Version 1.2
- http://gagius.co.uk/eliteforce.html: Seite mit Bombenentschärfungsmod, einem Masterserver und einem Masterserver Abfragetool.
- Interview mit den Entwicklern von EF 1 und 2
- https://www.stvef.org Schöne Seite mit Hintergrund-Infos zu EF, und auch einigen Mods.

Siehe auch

Star Trek Spiele

Zurück zur Games-Datenbank

From:
https://mwohlauer.d-n-s.name/wiki/ - **mwohlauer.d-n-s.name / www.mobile-infanterie.de**

Permanent link:
https://mwohlauer.d-n-s.name/wiki/doku.php?id=games:star_trek_-_voyager_eliteforce&rev=1561127188

Last update: **2019-06-21-16-26**