

Morrowind Mod Maker's Manual

Compiled by Edwardsmd
v2.0



Disclaimer

I am not in any way affiliated with Bethesda Softworks, ZeniMax Media, Microsoft, or any company or person who works for these companies. None of these companies in any way allowed or prevented me from completing this work, and most likely were unaware of it's production prior to it being provided over the Internet. Nor did any of these companies or anyone who works for them in any way provide any input or help me in compiling this work.

All words within are solely the product of myself. While much of the information within I learned through trial and error, a vast amount of it I gleaned from website forums and many various FAQs and tutorials provided freely by people within the gaming community. I do not provide or credit the source of specific information, next to any information within, as separate from that which I provide, as the manual would double in size. Therefore I take no credit whatsoever for anything contained within, only the compilation. All known sources of information will be listed separately from here.

This work is provided freely, and may be downloaded, printed, saved, or distributed to others through any means. However, you may not alter this work, nor profit in any way, if you distribute this work or provide it for distribution, without the express permission of myself and those listed under the Mojo Gratitudes section. This is their work too.

This work is currently version 2.0. I have taken pains to provide accuracy, but there is still some information within that is unknowingly incomplete, inaccurate, or false. I take no responsibility for anything any person does with their computer system due to the information contained within, nor any problems that may arise at any point in the future. Use this work at your own risk.

If you see information contained within that you originally provided and wish it removed or changed for accuracy, please contact myself and I will do so immediately.

-Ed

Table of Contents

Disclaimer.....	I
Table of Contents.....	II
Introduction.....	V
Mojo Gratitudes.....	VI
Abbreviations and Definitions.....	VIII
Shortkeys.....	X
Chapter 1 TESCS-What is it?.....	1
How a plug-in works.....	1
What you can do.....	1
What you cannot do.....	1
Version number of the Construction Set.....	2
What version of the Construction Set do I have?.....	2
What is an ESM/ESP/ESS file?.....	2
So then, what the heck are these nif, kif, tga, dds, bsa, etc. for?.....	3
Editing the Morrowind.ini file.....	3
Create an 'AllowYesToAll' button.....	4
Opening Multiple Construction Set windows.....	4
Forcing a maximum Frames per second (PFS).....	4
Enabling screen shots.....	4
Create a master file.....	5
Plug-ins being played currently.....	5
How do I load files?.....	5
If your mod will not run right because you now own Tribunal/Bloodmoon.....	6
I don't have Tribunal/Bloodmoon, but the mod I downloaded needs it/them to run!...	6
Can I download the TESCS, MW, TB, and/or BM from the internet?.....	6
I bought Morrowind, so it's legal for me to use a no-CD crack, right?.....	6
Should I copy the CS CD Data Files folder to my HD?.....	6
Evil dialog string messages!?!.....	6
Dependencies.....	7
ESPs dependent upon ESPs?.....	7
Evil GMSTs.....	8
Chapter 2 Installing/Playing Mods.....	9
The file is a 7ZP/RAR/ACE/ZIP! HELP!.....	9
Where do I put all these files?.....	10
How do I play this Mod?.....	11
Where's the Console?!.....	11
Console Functions.....	11
<i>Call the exterminator! Bugs everywhere!</i>	15
Can I play all 5-gigazillion mods I installed?.....	17
Chapter 3 Toolbar Description.....	18
The Toolbar.....	18
The Menus.....	18
File Menu.....	18
Edit Menu.....	21

View Menu.....	22
World Menu.....	24
Character Menu.....	27
Gameplay Menu.....	28
Help Menu.....	29
The Buttons.....	30
Data Files.....	30
Save.....	30
Preferences.....	30
Undo/Redo.....	30
Use Grid Snap.....	30
Use Angle Snap.....	30
Landscaping.....	30
Edit Cell Path Grid.....	30
Brightness.....	31
Fog.....	31
Dialog.....	31
Scripting.....	31
Sound.....	32
Chapter 4 Windows.....	33
The Objects Window.....	33
Activator.....	35
Apparatus.....	35
.....	Alchemy
35.....
Armor.....	36
Table 4.0 Armor Type Weight Allocation.....	37
Body Part.....	38
Book.....	39
Clothing.....	39
Container.....	40
Creatures.....	40
Door.....	42
Enchanting.....	42
Ingredient.....	43
Leveled Creature.....	43
Leveled Item.....	44
Light.....	44
Lockpick.....	45
Miscellaneous Item.....	45
NPC.....	45
Table 4.1 Fight Value Reactions.....	47
Table 4.2 Monetary Results Of Crimes.....	48
Probe.....	49
Repair Item.....	49
Spellmaking.....	49
Static.....	51

Weapon.....	51
The Render Window.....	52
The Preview Window.....	54
The Cells View Window.....	54
The Race Window.....	55
The Skill Window.....	56
The Class Window.....	57
The Birthsign Window.....	58
The Landscaping Edit Window.....	59
The Edit Cell Path Grid Window.....	62
The Faction Window.....	63
The Dialog Window.....	64
How Dialog Works.....	66
Speaker Conditions.....	67
Table 4-3: Dialog Greetings Sublevels.....	75
Dialog Review.....	77
The Scripting Window.....	78
Making Scripts.....	79
How Scripts Work.....	81
The Commands.....	82
Syntax.....	84
The Functions.....	86
Global and Hard Coded Variables.....	126
Character Generation.....	129
The Magic Effects Window.....	132
The Sound Window.....	133
The Sound Generator Window.....	133
Chapter 5 Saving and Testing your Mod.....	135
Creating a mod.....	135
Doubling?!? I usually call it stupid, @&^+&#!, &*\$%, @^&*\$#%*!!!.....	136
Cleaning the Unclean.....	137
Respecting things other people make.....	139
Chapter 6 Utilities.....	141
BSA Browser.....	141
BookGen.....	141
DDS Converter.....	141
Enchanted Editor.....	141
ESP Cleaner.....	141
FPS Optimizer.....	141
GenMods.....	141
GMST Cleaner.....	141
Landscape Generator.....	141
Leveled List Merge.....	141
Morrowind Clothing Catalog.....	141
MWEdit.....	142
Morrowind Script Generator.....	142
NifDuplicator.....	142

NifTexture.....	142
OverUnity.....	142
The Complete Guide to Tamriel Lore.....	142
TESFiles.....	142
TES Item Index.....	142
TESAME.....	142
TESDTK.....	142
TESFaith.....	142
TESMU.....	143
TESPCD.....	143
TESTool.....	143
Difficulty slider vs. tweak mods.....	143
Appendix A References and Tutorials.....	144
Appendix B Tables.....	145
Race List.....	145
Spell Effects List.....	145
Scripting/Command Function List.....	146
Game Settings List.....	164
Appendix C Websites.....	173
Appendix D 3D Meshes and Textures.....	176
Appendix E Contacting.....	178
Index.....	179

Introduction

Wow. What a trip it's been. Here we are several years since Morrowind has been released, and people are still playing, and still modding. Few games have as large and devoted a community, or as great. My own journey with Morrowind has been just fantastic. Prior to Morrowind's release, I didn't even know what a 'mod' was. Being online meant checking email, nothing more. While I consider myself an avid gamer, I tend to limit my interests to RPGs, and a couple strategy games. I knew of Arena only because a friend had it when I was young, though I didn't have a computer then (my parents relented and bought me a calculator for school work). I bought Daggerfall when it came out, but I was still on my first computer at that point, it was a bit old, and the game didn't run well. I also lived in a different country at that point in my life, and had no reliable internet connection. Still, despite the CTDs, slow gameplay, and other problems, I loved DF (hey, it was my first computer, I was still learning). Alas, during the move back to the US, it disappeared. I contented myself with Masters of Orion, Might & Magic 6, and Baldur's Gate II (with Throne of Baal of course). Knowing Morrowind would be coming out, I began saving money. Solely due to this I actually bought a couple game magazines that had articles about it. Then finally picked up a copy of the game. I was hooked from the get go. I played the heck out of it. But what was this Construction Set that came with it? I tried messing with it, but I was so lost, so mostly gave up. I was having fun playing. By now of course I was online, and actually once in a while did more than check email. One day I did a search for Morrowind, just to see what turned up. Well, a lot of links got returned, but I didn't click any, until I saw a link for something called Morrowindfiles. What can I say? That place opened a new world to me.

I discovered things called 'mods'. I even downloaded a couple and learned to install them. I was in heaven. Giddy with my discovery I reopened the Construction Set and gave it another whirl. No luck, I still had no clue. So I continued my occasional perusal of Morrowindfiles. I noticed it had something called a 'forum', but never bothered to check it out. Then the unimaginable. Morrowindfiles died. This forced me to look elsewhere, and in the process I discovered the vast community of websites devoted to Morrowind. I ended up registering at several, and began posting on a few forums in my quest to understand this thing called the Construction Set. I gleaned a great deal of info this way, which only fueled my descent into madness known as modding. I began making all kinds of little mods for myself, none of any real value, just fun stuff for me. And as I became a part of the community more and more, I learned more and more, which enriched the entertainment value of Morrowind enormously. Like many though, I was eternally frustrated at the lack of documentation and explanation for how to use the Construction Set. Looking around at all the notes I made, and all the post-it notes stuck everywhere, I decided to just make something for myself to use.

Thus I created the first version of this work, though in a very simplified form. I provided a copy to the community, but was dissatisfied with it. It wasn't very good. So I set about making something comprehensive, and released it. Since then I got around to actually creating a mod and releasing it, and hope people enjoy it. But using my own work as a reference, I began to realize how incomplete the manual I had compiled was. I decided to update it. This has proven easier said than done. I didn't want to just add more content, I wanted it to be a better manual, not just for the newb, but for the expert also. I want anyone to be able to not only learn how to use the CS with this manual, but also be able to use it as a reference as they work. I hope this version has been able to accomplish this.

Some things to keep in mind about this manual. I tried hard to include everything about the CS in it. Impossible, I don't know everything, and I learn something new every day. While I would love to do something entirely comprehensive, this work would be HUGE were I to go that route. Ever tried to find that one tidbit of info you need in some 500 page reference manual? Hateful stuff. So, I have sacrificed some to enable this manual to be a simple, but complete work. You will not find any tutorials in here on how to make a quest, or how to script, or the best way of how to do 'this' or 'that'. But what you will find is the info you need to understand how to use the various functions to make that quest. How to use the scripting engine to make a script, the dialog engine to make dialog (yes, I included dialog this time), or the landscape engine to make landscape. I won't tell you the best way to create anything, but I will explain what all the menus and buttons do. What you can do when you open this or that window. And yes, I will offer little tips now and then, usually to help you avoid common pitfalls as you work with the CS.

You will most likely still find some inaccuracies, this is inevitable. I'm human, I make mistakes (my wife is real good at finding these). But I have tried my best. My only hope in compiling this work is to produce something that people can use to help them mod. This shouldn't be your only resource to help you as you work on a mod, but I hope it's a valuable one.

With the advent of TES IV: Oblivion on the horizon, I doubt I will do another update. Besides, I'm busy modding, and still playing Morrowind. The meager amount of info known so far for TES IV's Construction Set sounds intriguing at the very least. For myself, I can't wait to get my hands on it. I already have ideas bouncing around inside my brain, and who knows, maybe I'll end up putting together a manual for it too.

Now then, *let's get modding!*

-Ed
17 June 2005

Mojo Gratitudes

The following people have in some way contributed to the consolidation of information presented here. Mostly by finding or figuring out said information, and either posting it within a forum somewhere or providing it in some way for distribution for all. I know I will be missing many people, but I will list all those I can that I know of. This list is presented in no particular order.

- Nether Void - His original compilation, the “Morrowind Mod Makers Bible” was invaluable. It was the inspiration to further refine and compile a comprehensive reference guide to the TESCS.
- Dragonsong - For the great tutorials.
- GhanBuriGhan - His Scripting For Dummies is essential.
- Dave Humphries - His website is packed with information and goodies.
- BigChief - a personal thanks for his help in learning the CS.
- Vaulimere
- Traven
- Martini60 - more personal thanks for his help in learning scripting.
- Ziegfelding
- Blith Erring Idio’ - untold kudos for his website Gamers Roam, and allowing me to be an active member of the community.
- Mysteria
- SecretChimp
- thefunky1
- Morgoth666
- Sapper
- Sorcerer
- Emma - she sets a standard for plug-in creation.
- The Other Felix - for pushing the boundaries with scripting.
- Grumpy - for his scripting work with companions.
- SquelchyUnderfoot - for sharing his extensive computer knowledge.
- Niyt Owl
- Kathykitten
- EsnRedshirt
- Smudge
- Saber
- Rukinea - For good tutorials.
- Scarabus
- Olsor
- Snograth
- ThebigMuh
- Shoujo

[Table of Contents](#)

- Riiak
- Rhyek
- Horatio
- Ragnar_GD
- Nazz
- Eldar Mayiere
- Striker
- B
- Shadowsong
- Raptormeat
- MadMax_001
- Iudas
- maxpublic
- Wakim
- Ldones
- Sheikizza
- CrisY
- Baphomet - An endless source of help on forums
- Klinn - Another great source of information on the forums
- Athane Khagan
- Calislahn - Her Dialog tutorial helped me the most in understanding it.
- Kawaii
- Emnesharra
- ThreadWhisperer
- ManaUser
- MistyMoon
- Canadian Ice
- Katheryn
- JOG
- Srikandi - Her dialog tutorial is extremely useful.
- Wyre
- OldCow69

While others should probably be listed, I do not know who they are, and of those listed above I only added comments to several because I am aware of the amount of information, or a specific resource, they have provided to the mod community. If someone feels snubbed by this, my apologies, that was not my intent. Contact me and I will endeavor to give all due credit to the best of my ability.

Abbreviations and Definitions

3D Studio Max - Professional program used to create the 3D objects of the game. Very expensive. Only versions 3, 4, or 5 can be effectively be used for Morrowind. Newer versions may not work as well.

.7z - A compressed file, like a .zip file.

.ace - A compressed file, like a .zip file.

Acrobat - Program used to create .pdf files. A tad bit expensive.

Acrobat Reader - A free program that can be used to view any .pdf file.

Adobe - Company that created PhotoShop, Acrobat, and Acrobat Viewer.

Attributes - The PC's Strength, Intelligence, Willpower, Agility, Speed, Endurance, Personality, Luck, Fatigue, Magicka, and Health.

.bmp - Bitmap file. An image file. Usually pretty big in kb.

.bsa File that holds all the meshes, textures, and icon files used by Morrowind (and Tribunal & Bloodmoon). Can also be used with an .esp file.

CS (or TESCS) - Construction Set. The reason you're reading this.

CTD - Crash To Desktop. When you're playing Morrowind and it suddenly stops, disappears, and is replaced by Windows, that's a Crash To Desktop.

d-click, d-click'd - Double click.

Data Files - The subfolder in your Morrowind directory that holds most of the data your game needs to play. Also holds the .esp files. A plug-in will not be shown on the start screen of Morrowind to load, nor be accessible to the TESCS, unless it is in this folder.

.dds - Direct Draw Surface file. An image file just like a .bmp, but not as big in kb, and with better compression. Allow use of alpha channels and mipmaps.

Dependent - Plug-in files are dependent upon one or more master files. You cannot load a plug-in in the CS, nor in the game, without loading the master files it is dependent upon.

Discreet - Company that owns and sells 3D Studio Max.

Divine Marker - Marker placed in the game to denote the point the PC is taken to when they teleport to an imperial shrine, via spell.

Door Marker - Marker placed in the game to denote the point the PC is taken to when they exit from a door.

Duplicate - You can duplicate any reference in the Render Window and create an exact copy.

.esm - Elder Scrolls Master file. You cannot affect this file in any way.

.esp - Elder Scrolls Plug-in file. All changes made with the CS are stored in this file.

.ess - Elder Scrolls Save file. These are your saved game files.

Expansions - Tribunal and Bloodmoon are the two official expansions available to the Morrowind game.

FPS - Frames Per Second. Your computer redraws the screen constantly. Each time it does this is referred to as a 'frame'. The more it has to draw the fewer frames per second. Less than 30 FPS is considered noticeable by the human eye.

GIMP - A free software program used to create 'image', or 'art', files (i.e. textures).

GMST - Game Setting.

HD or HDD - Hard Drive, or Hard Disk Drive.

[Table of Contents](#)

[Table of Contents](#)

ID - Identification (name of an object or thing in the CS). Not necessarily the *Name* of an object, but the *ID name* the CS keeps track of an object by.

.kif - File associated with a .nif file that has animation in it.

l-click - Left-click.

Leveled List - A list created with creatures or treasure. It is accessed by the game to provide a creature to fight appropriate to the PC's level, or distribute treasure appropriate to the PC's level.

Line of Sight - The unobscured line from an object to another object. If an object could *see* another object, then it has Line of Sight to that object.

Meshes - The 3-dimensional wireframe form of an object.

MilkShape - A cheap program used to create meshes. Not as capable as 3D Studio Max.

Mod - A plug-in, to include all files necessary to play it.

.nif - NetImmerse File. A mesh file. Used by Morrowind.

Northmarker - Object placed in an interior cell to denote which way is north.

NPC - Non Player Character, every character in the game except the one playing.

Object - Anything that can be placed in the game, is an object, and treated as such by the game.

Paint Shop Pro - Software used to create 'image', or 'art', files. Not too expensive.

PC - Player Character, the character used by the one playing the game.

.pdf - Portable Data File. This type of file can be opened and viewed by anyone on a pc or mac computer with Adobe Acrobat Viewer.

PhotoShop - Software used to create 'image', or 'art', files. A bit expensive.

r-click - Right-click.

.rar - A compressed file, like a .zip file.

readme.txt - A simple text file that explains what a file is for, what it does, how to install it, and what to expect. Basically everything about the file. Should be included with any mod you create, upload, or download.

Reference - Any time you place an object in the game you create a *reference* of that object.

Script, Scripting - Manually written code to execute, or augment, an effect in the game (often hidden from the player, though it's effects or results may not be).

Skills - Statistical actions you can take to accomplish something in the game.

Spell - A magical effect cast by actors/PC composed of spell effects.

Spell Effects - Spell effects are used to create spells that are used in the game world.

Skin - The texture file that is 'wrapped' around a mesh to create an object.

Skinning - The process of altering and fitting a texture onto a wireframe to create an object.

Stats - The PC's defining statistical abilities (attributes and skills).

TC - Total Conversion. When someone completely removes everything from Morrowind (all cells, landscape, references, etc) except the game engine, and creates a new "game" from scratch as an .esm file.

Teleport Marker - Marker placed in the game to denote the point the PC is taken to when they 'arrive' via an NPC teleportation service.

Temple Marker - Marker placed in the game to denote the point the PC is taken to when they teleport to a tribunal shrine, via spell.

TES - The Elder Scrolls.

TESCS (or CS) - The Elders Scrolls Construction Set.

Textures - An 'image' file used to 'skin' a mesh to create an object.

.tga - Targa Truevision Graphics file. An image file, like a .bmp.

Toggle - Usually a key, or combination of keys, that produces an affect, or stops it, by 'turning it on' or 'off' with repeated keystrokes.

Travel Marker - Marker placed in the game to denote the point the PC is taken to when they finish 'fast' traveling (i.e. silt strider or boat).

.zip - A compressed file.

Shortkeys

Many shortkeys in the CS do not need a key combination to work (i.e. you do not need to hit ctrl + 'key' to accomplish something). Just hit the shortkey to toggle or activate a command or response within the CS.

In the **Render Window**:

- A - Toggles on/off brightness with the Render Window
- B - Toggles on/off yellow border lines delineating cells
- C - Centers your viewpoint upon an object at ground level, facing north
- D - Deselects your viewpoint from an object
- F - Drops, or makes fall, an object to the ground or object underneath it
- H - Toggles on/off the Landscape Editor
- L - Toggles on/off viewing the radius of light source objects
- T - Centers your viewpoint above a selected object
- S - When held, click on an object to resize it
- V - When held, allows you to zoom in or out by moving your mouse
- W - Toggle on/off wireframe mode
- X - When held, allows you to move an object with your mouse on the X-axis
- Y - When held, allows you to move an object with your mouse on the Y-axis
- Z - When held, allows you to move an object with your mouse on the Z-axis
- F4 - Shows collision detection of all objects within the Render Window
- 'Shift' - When held, allows you to rotate, or spin, your viewpoint 360 degrees with your mouse (if an object is selected, you will rotate around that object)
- 'Spacebar' - When held, allows you to change your viewpoint upon the XY-plane (sideways), or the Z-plane (up and down) with you mouse
- 'mousewheel' - allows you to zoom in or out, if you have one

In the **Preview Window**:

- A - Pans the view left
- D - Pans the view right
- S - Pans the view down
- W - Pans the view up
- Insert - Rotates the object counterclockwise on it's 'x' axis
- Home - Rotates the object clockwise on it's 'x' axis
- Up Arrow - Rotates the object up on it's 'y' axis
- Down Arrow - Rotates the object down on it's 'y' axis
- Left Arrow - Rotates the object left on it's 'z' axis
- Right Arrow - Rotates the object right on it's 'z' axis
- Page Up - Zooms the view in
- Page Down - Zooms the view out

***Note:** For the Preview Window, I have presumptively assigned which is the 'x' and 'y' axis merely to define the key's assignment to an axis. I do not know if the axis I state as 'x' or 'y' is correct.*

^T[Table of Contents](#)

^T[Table of Contents](#)

Chapter 1 TESCS – What is it?

“The Elder Scrolls Construction Set allows you to edit and create any data for use with The Elder Scrolls 3: Morrowind.”

-from The Elder Scrolls Construction Set help file

That pretty much sums it up. With the CS you can create new items, NPCs, spells, races, birthsigns, towns, dungeons, classes, factions (i.e. guilds), quests, or alter what's provided with the game. There are still some things you cannot do with the CS, and there are some things you *should not* do with it. When working with the CS, keep in mind the reason it was provided: to *extend* the life of Morrowind as a product for *sale*. Period. The designers of the game may feel some sense of altruism in releasing the CS, but you're praying for a jackalope sighting if you attach the same to those who published the game. They did it to increase *profit*. With this in mind realize there are built in limitations. So when you want to cry because you can't create your own .esm file (we'll get to this), or wonder why you can't create new [Skills](#), [Attributes](#), or [Spell Effects](#), new or different 'bodies' for [Races](#) (3D Studio Max? What's that?), and curse and kick at the shortsightedness of those who created/limited the [Scripting](#) commands and functions, relax. Just remember why, and say to yourself, “If I were a god of Death and Destruction!!!”... but you're not, so forget about it. Enjoy the game, and have fun modding.

How a plug-in works

When you work on your plug-in, or mod, you are not actually changing anything in the `morrowind.esm` file. You are making changes stored as an .esp file. When the game is loaded, the CS loads an .esm (or more than one), then loads any plug-ins. The CS then makes changes upon the active plug-in. What this means is, you cannot do anything that can ruin the game itself. If a plug-in causes problems, just exit the game, open up the Data Files menu, uncheck the plug-in(s) causing problems, and click Play. Master files cannot be affected in any way.

What you can do with the TESCS

What you are able to do is quite a lot, but can be summarized this way:

- You can alter almost everything within the game.
- You can add new objects (weapons, spells, races, land, towns, dungeons, etc) by re-using what is provided within the CS, or creating new ones based off of what is provided.
- You can modify objects within the CS to do something different.
- You can modify the way the game world operates and how the player interacts with it.
- Create and/or modify an .esp file.
- You can wreck the game entirely rendering it completely useless by messing with things you don't know about (Which hopefully this manual can prevent).

What you cannot do with the TESCS

What you cannot accomplish is actually quite small, but very absolute. This is done to protect the stability of the game, and enable mods produced by different people with different ideas to

work together if installed, and loaded in a game at the same time. Also to protect the profit line of the game. You cannot:

- Create new [meshes](#) or [skins](#) for items or races in the game (see [Appendix D](#)).
- Create [spell effects](#) used to create [spells](#) within the CS.
- Create [skills](#) or [attributes](#).
- Make changes to the physics engine (though you can alter some aspects of how it operates).
- Create or alter an `.esm` file.
- Create or alter an `.ess` file.
- Become a god! (er, at least not in reality)

***Note:** [Scripting](#) enables you to accomplish things the designers may not have thought of, or get around inherent limitations of the game. This was intentional, but scripting also has a great many limitations to prevent abuse, maintain stability, and to keep the spirit of the game in line with what is stated in the above bullets.*

Version Number of the Construction Set.

Version 1.0 comes with the Morrowind CD. If you install either Tribunal or Bloodmoon it will be upgraded to v1.3. Note that there are new different functions included with each, however both upgrade the CS to v1.3.

What version of the Construction Set do I have?

If you have Tribunal or Bloodmoon you have v1.3. To see the version used to create any mod open the Data Files window and highlight any `.esp` file. You will see the version number on the right hand side.

What is an ESM/ESP/ESS file?

These are the files that hold all the information, or data, the game uses to run. In simple terms:

- **.ESM** (Elder Scrolls Master) files are the **master files** that contain all the data to play the game, everything. The `morrowind.exe` file holds the code on how the game functions. This keeps things simple, and also why unlike other games you don't need to install a 2.0 GB file on your hard drive to play. You cannot alter, change, modify, or delete anything in an `.esm` file with the CS. Even if you try. Why did Bethesda do that? So we can't screw the game up and spend all our time on the phone with Bethesda's customer service bitching and complaining about their stupid game that we screwed up.
- **.ESP** (Elder Scrolls Plug-in) files are the **plug-in files** that we create when we save changes we have made to the game. We can alter, modify, add, and delete these files to our late-night, bleary-eyed, caffeine-induced heart's content. Then post it on the Internet for others to try out. You can even do this to someone else's `.esp` file you've downloaded from the Internet. Some few have done this to modify and improve a good mod that just needed a few tweaks to

make it a great mod. Of course, some post dirty/buggy mods too. Thankfully, this is somewhat rare.

- **.ESS** (Elder Scrolls Save) files are your **saved game files**. Lots of people like to make saved game editors to cheat and boost their character's abilities for games that are published, especially RPGs. I have even seen some for Morrowind. Why? With the CS you can already do this outside of game play, or with the console ingame. You shouldn't have any reason to mess with this file.

***Note:** It is possible to change the `.ess` file extension to `.esp`, then use the CS to view the contents of the file by clicking the Details button in the Data Files window (see [Viewing a Save file](#)).*

So then, what the heck are these nif, kif, tga, dds, bsa, etc. for?

A simple statement explaining each.

- .bmp** - An art file. Most people are familiar with them. They are commonly used as a texture or 'skin' for meshes.
- .bsa** - This file is named after the `.esm` it belongs with (i.e. `Morrowind.esm` and `Morrowind.bsa`). It stores all the object files used by the `.esm` when playing. This file can be huge. There are utilities created by several skilled modders that enable one to open the `.bsa` and view, extract, or add files to it (see [Chapter 6](#)).
- .dds** - Another type of art file used as a texture or 'skin' for a mesh. Compresses well, and allows use of alpha channels (for transparency effects) and mipmaps.
- .fnt** - This file is a font.
- .ico** - This file is an icon.
- .jpg** - Yes, believe it or not, some people use these as skins for meshes. I have no idea how well they work (if at all), but the game seems to prefer `.dds` texture files.
- .kif** - This file comes along with a `.nif` file for animation purposes.
- .mp3** - This is a compressed audio file that provides music for the game.
- .nif** - This file is the mesh for all physical objects in the game. Texture files are associated/linked with it that provide the 'skin'. More than one texture file may be linked to provide the skin. Most texture files are either `.bmp` (rarely) or `.dds` (almost all). If an object has animation then a `.kif` file usually will be associated with a `.nif`. Some `.nif` files may also have a second `.nif` file that holds additional data for animation (such as an activator). These usually have the same name as the `.nif` file it is associated with, but the name will start with an 'X'.
- .tga** - This is an art file also, smaller than a `.bmp`, but not as small as a `.dds` file. Also allows use of alpha channels.
- .txt** - A simple text file.
- .wav** - An audio file. Most sounds in the game are this type of file.

Editing the `Morrowind.ini` file.

This specific file provides data for the game to start up and run. I don't suggest you mess with this file unless you know *exactly* what you are doing. However, there are tweaks you can do to this file to improve or fix your game if it has trouble running properly. A few mods also force

you to make minor changes to it for them to run properly. If you wish to make changes to this file, ***make a back up copy first***. If you screw it up or your changes cause problems you can always replace it with an unchanged copy.

Create an 'AllowYesToAll' button.

You can also modify your `morrowind.ini` file to create just one button to click instead of clicking a bunch (make a back up of your `morrowind.ini` file first). Open up the file and at the bottom of the general section, on a separate line by itself add; 'AllowYesToAll=1'. When you start up the CS you can now hit the Cancel button for a 'Yes to All' option. Do this and skip the finger exercises.

***Note:** When starting the game, a 'Yes to all' button now appears when an error is generated.*

Opening Multiple Construction Set windows.

Yes, you can do this. Simply open up the `morrowind.ini` file and add 'AllowMultipleEditors=1' somewhere in the general section on a line by itself. Now you can open multiple windows of the CS at one time.

***Note:** Be advised, the CS is not the most stable program ever created. I do not recommend having extra multiple Construction Sets open for more than reference to a master file's or mod's contents. Going crazy with this has resulted in crashing some of the CS windows on my system. But that may just be me.*

Forcing a maximum Frames per Second (FPS).

If you find your game having trouble attaining high FPS, you can lower the maximum amount of FPS the game will run. This will keep your system from choking trying to turn out 240 FPS, every second. Open your `Morrowind.ini` file again, and near the top you will see a line that reads `Max FPS=240`. Change the number after the '=' to lower, say 60, or even 40. This will save your system resources. Even if you are so lucky to have a Cray sitting in your computer dungeon haven, your eye won't track more than about 30 PFS, so 60 FPS is reasonable.

Enabling screen shots.

Dying to get some screenies posted of your mod? Open up the `Morrowind.ini`, about the middle of the general section near the top, you will find `Screen Shot Enable=0`. Change the '0' to a '1'. Now when playing ingame, when you want to take a screen shot, just tap the `Print Screen` button. A screen shot will be created in your `Morrowind` folder on your HD, in `.bmp` format. See the line that says, `Screen Shot Base Name=ScreenShot?` Change 'ScreenShot' to whatever you want, and all your screenies will be named that. See the line below that that reads `Screen Shot Index=10?` That's the index number of the last screen shot you took, or how many you've currently taken up to now (I had last taken 10). You can also create a separate folder, type in the path here, and all screenies will now be stored there. Like this:

```
Screen Shot Base Name=C:\Program Files\Bethesda Softworks\My  
Screenshots\ScreenShot
```

Create a master file.

Turn an .esp file into an .esm file? This can be done. But rarely should it be done. I will not expound upon this, except to say you had darn tooting better know what you are doing here. A good rule of thumb is, if you change ANYTHING in an .esm file, do not do this. You will most likely be screwing something up. If your mod does nothing more than add objects, then probably this would be ok. However, keep in mind the whole point of an .esp is so you can make changes to the game *without screwing something up*. Now ask yourself, "Do I need to make this an .esm?" If you have to think about it, you don't. That said, this will not create a master file for that TC you want to make. This method will create an .esm that will be dependent upon any master file the .esp is dependent upon (sorry charlie).

Still set on it? Ok, first open your Morrowind.ini file. Scroll down to near the bottom until you see [WhoCanMerge]. Below that name put in the name of your computer, followed by '=1'. It should look like this:

```
[WhoCanMerge]
yourcomputername=1
```

To find your computer name open up the Program Flow.txt file in your Morrowind folder. The second line will list it. Add you computer name as directed above. Save the Morrowind.ini file (of course you backed it up). Now then, open the CS, mark your .esp as per normal, set it as active. Click Merge to Master which is no longer grayed out, click ok. That's it. Your plug-in is now an .esm file. Oh, and you can't change it back with the CS either. Shadowsong has a good tutorial that covers many of the issues about converting a plug-in to a master file (see [Appendix C](#)).

Moving dialog responses with filter active. You may see this line in the Morrowind.ini file: CanMoveInfosWhileFiltered=0. As you may suspect, changing the 0 to a 1 will allow you to fully mess with dialog while filtered. Don't do it, besides dirtying up your mod, you could possibly cause real problems.

See travel methods on your map while playing. Find the line Show Travel Lines=0 in your Morrowind.ini file, change the 0 to a 1. Now when playing you can see the travel lines on your map for Guild Guide travel, Silt Strider travel, and Boat travel. Each will be a different color.

Plug-ins being played currently.

Wonder how to get a list of all 200+ mods you are currently playing with, without having to write them down by hand? Open the Morrowind.ini file (again), and at the very bottom you will see all plug-ins last played under the heading [GameFiles]. Easy huh.

How do I load files?

When you load files into the CS it works in a similar way to starting a game with a plug-in. Click any .esp files you want loaded, you may click more than one. Highlight the one you want *active*. Any changes made will be saved to the *active* .esp. Changes will be denoted with an asterisk (*) next to the ID of any object or cell. Now, you need to keep something in mind here; if you make a *change* to an object from a loaded but not *active* .esp, you might just have made **your** .esp *dependent* upon that *non-active* .esp file. I'll explain this more under [Dependencies](#). When you click the save button, the active .esp will become dependent upon any loaded .esm files. Keep this in mind when choosing what you want to load and base your .esp off of (again,

see Dependencies). A cautionary note here, you do not have to place a check next to an .esm when loading an .esp, the game will load any .esm files an .esp is dependent on. However, I have generated errors when doing this at times (no clue why, and it doesn't happen often or with all plug-ins). I suggest you always mark what you want to load. There is no need to make an .esm active, the active button is only for .esp files. The .esm files always load first, then .esp files that are marked. The .esp files are listed in the Data Files window from top to bottom, by date (oldest date at the top, newest at the bottom). The date stamp on an .esp is when last Saved.

***Note:** When .esp files load, they are loaded oldest first to newest last. No big deal really in the CS, but when you want to play a game this can be a big deal.*

If your mod generates errors when loading because you now own Tribunal/Bloodmoon:

This is because any mod created has the version number of the CS associated with it. Tribunal and Bloodmoon update the CS version number. Just ignore the error statements, or load the mod with Tribunal and/or Bloodmoon in the CS and click save. This will update the mod version number and you won't get those any more.

***Note:** If you choose to update a mod to a new version number, be aware you will add the 'Evil GMSTs' to the mod. See Evil GMSTs below for more info.*

I don't have Tribunal/Bloodmoon, but the mod I downloaded needs it/them to run!

Sucks to be you. Go buy them.

Can I download the TESCS, Morrowind, Tribunal, and/or Bloodmoon from the internet?

No. Doing such a thing would be *illegal*. Besides, Bethesda deserves a few bucks for creating such a great game.

I bought Morrowind, so it's legal for me to use a no-CD crack, right?

Wrong. Use of a no-CD crack is illegal, period.

Should I copy the Data Files folder on the Construction Set CD to my Morrowind Folder?

This is subjective, and entirely up to you. The game by default looks for object files on your HD first, so if something is modified, it uses it. Then it will go to the appropriate .bsa file and use what it finds there. So it won't hurt anything if you do this, I personally don't see any reason for it.

***Note:** The files used for Tribunal and Bloodmoon can be found on those respective CDs.*

Evil Dialog string changes!?!

Ever get those stupid messages when loading up the CS after you installed Tribunal and/or Bloodmoon telling you some dialog string has been changed. Sick and tired of clicking 'yes' over and over. Ever hit 'no' because you didn't know better. What to do about it? The short answer? C4 and a blasting cap. No, no, no, no... Just ignore them, and keep clicking 'yes' to continue. Why does this happen? In case a dialog topic has changes in an .esm or .esp different from the morrowind.esm file (TB and BM do). Ignore them, or read the next paragraph.

Dependencies

All `.esp` files are dependent upon one or more `.esm` files. This is because an `.esp` doesn't store the information about objects in the game, it stores *changes made* to any objects in the game. When you load an `.esp`, all `.esm` files it is *dependent* upon will be loaded also, whether or not you marked them for loading (see [How do I load files?](#) above). You cannot prevent this. When you click the Save button, your active `.esp` file becomes *dependent* upon any `.esm` files loaded. If you look you will see that both Tribunal and Bloodmoon `.esm` files are dependent upon the Morrowind `.esm` file (assuming you have them). Do you want the mod you're making dependent upon Tribunal and/or Bloodmoon? Well, if you are using an object of the game that came with one of these expansion packs, yes, load them up and make your `.esp` file dependent upon them. But if not, don't whimsically load them along with your `.esp`. There are two main reasons for this; the first being that it will limit who can play your mod. You're cutting out anyone who chooses not to buy one or both expansion packs if you add one to your mod. The second being what are referred to as the Evil Game Settings (or Evil GMSTs for short). I'll explain this later on, but believe me, they can be a pain to get rid of if you don't want them in your mod. But they can be removed with a utility (see [Chapter 6](#)).

.ESP files dependent upon .esp files. Ok, another potential problem: `.esp` files that become dependent upon another `.esp`. It can happen, and does. New modders tend to do this and don't even realize it. I'll explain as best I can. Let's say you load another modder's plug-in because they have new item models you like; such as trees, hair, heads, weapons, etc. Your plug-in is the active one. Now:

1. You **rename** the object for use in your plug-in. This is the ideal way to use an object created by another. Your plug-in *won't* be dependent on the other's `.esp`, but it might be dependent upon textures or meshes if created by the modder. This is because you're creating a new object based off of another. If this is done you should get permission to use the author's meshes/textures first, and then you'll have to package them with your mod. Otherwise no one will be able to use your plug-in unless they already have the same plug-in's meshes and textures you borrowed the object from.
2. You **don't rename** an object, you just click on an object made by someone else and drop it in the game world somewhere. Whoa! You just made your plug-in *dependent* upon another modder's plug-in. Why? Because *your* plug-in only tells the game where to put a reference for the object. The *other's* plug-in tells the game all the data about the object. Now, to use your plug-in, a gamer must also have the other plug-in installed and running when they play. Not only that, but if you change any of the data about said object, you may just break the other modder's plug-in if it's important to a quest or some such. Be smart, rename objects you want to use for your own mod.

You'll see more about renaming objects later on.

Evil GMSTs (*cough, hack, wheeze...*)

What are they? If you have Tribunal and/or Bloodmoon installed, you have them. Simply put, if you have one or both expansions installed, every time you click the Save button on your mod, the newer GMSTs from the expansion packs get added to your mod. Not a big deal if your mod has them as dependent, but they can screw with a mod if not. Many people just don't care for

them. How do you know if you have them? If you have Tribunal, and create a mod with just Morrowind, you'll have the Tribunal extra GMSTs. However, if you have Tribunal (but not Bloodmoon), and make your mod dependent upon it, that's ok. The GMSTs are supposed to be there. Same goes for Bloodmoon; if you have only Bloodmoon, and create a mod without it, the CS will put the extra GMSTs that came with it in your mod. But if the mod is dependent upon Bloodmoon, you're ok, they should be there. If you have both installed, but don't have your mod dependent one or both of them, you'll get the respective GMSTs in your mod. How to get rid of them? First you have to have a utility that can open up your `.esp` and identify them. For the sake of convenience, I will explain how with TES Advanced Mod Editor (or TESAME, see [Chapter 6](#)). Start up TESMAE, open up your `.esp` file. Look in the left column for any object marked as 'GMST'. R-click, or highlight it, and hit the spacebar so that it is a black color. Once you have found and marked them all, hit the delete button and get rid of them. Save your `.esp` and you're done. Be advised, if you open your mod up with the CS again and hit save, all the GMSTs will be loaded into your `.esp` again and you'll have to repeat this process. This is best saved as the last thing you do with your mod, if it's necessary.

But let's say you changed one or more of the GMSTs to a value you want for your mod. Fine, just be careful not to delete the change you made. I won't go into a lot of detail on this. For more information, I personally recommend you either check out the official forums, or a forum of your choosing. A simple search should bring up plenty of GMST topics to read up on. Argent also has a good explanation of GMSTs at his website (see [Appendix C](#))

Chapter 2 Installing/Playing Mods

Where do you get all the best mods? You can probably get them from the same place you got this reference manual, or just check out [Appendix C](#). Once you have the mod you most likely will have take steps to unzip it, possibly place files where they belong, and then you can play it. Maybe. First, let's figure out what file format the mod you downloaded is in.

The file is a 7ZP/RAR/ACE/ZIP! HELP!

Ah yes, I remember the first time I downloaded a file, saw the [.rar](#) extension, and said to myself, "What the *#&^!! is that!!!" Well, fear not o' ignorant one. It's just a different form of file compression, as is an [.ace](#) format. Open either one and if you have a utility that can open them, you're in business. If Windows brings up a little window asking you to choose the program to use with this file, you need a utility or program to unzip these mods. Almost any compression utility will open a [.zip](#) file, some will open a [.rar](#), and others will open an [.ace](#). Here is a breakdown for you, but I make no suggestions, I'll let you decide what's best for you. The websites to find these can be found in [Appendix B](#).

These will open a [.zip](#) file:

- WinRar
- ZipGenius
- WinZip
- PKZip
- 7zip

These will open a [.rar](#) file:

- WinRar
- ZipGenius
- 7zip

These will open an [.ace](#) file:

- WinAce
- ZipGenius
- 7zip

These will open a [.7z](#) file:

- 7zip
- ZipGenius

Keep in mind: *You get what you pay for* is a good rule of thumb here. I tried several different 'free' utilities that kept screwing files up when I tried to extract them. Maybe it was the program, maybe it was just my system. I have found those listed above to be reliable though.

Where to put all these files?

[Table of Contents](#)

Most people include a [readme.txt](#) file with their mod (*and a pox upon those that don't!*), so **read it**, that's what it's for. They should tell you where to put the files that are included. In general, the files should always go somewhere within the [Data Files](#) subfolder within the Morrowind folder (most of the time). Some modders will produce a .zip file that will *self-extract*. This means all you have to do is open or d-click it. The file will do the rest and you're done. Most files you will be prompted to choose where to extract the files. The [readme.txt](#) provided is the source for this info, but most likely it will be the [Data Files](#) subfolder. Most compression utilities have the ability to open a [readme.txt](#) while still compressed. Just open the compressed file and look for the [readme.txt](#). Double click it, and it should open for your reading pleasure. If you are leery of unzipping straight into you Morrowind subfolders (and I don't blame you), just extract to a temporary folder to examine the contents, and then you can manually move them over. This is my preferred method. If you don't know what you are doing, or are not conversant with file placement though, you could mess something up. Again, I **highly** recommend you do what the [readme.txt](#) tells you to do.

No readme.txt included? What sick bastard uploaded this crap?! Delete it! If you don't know what the different files are that Morrowind uses, and where to put them, you don't know enough to be messing around like this. You might accidentally replace a needed file, or put something in the wrong place. I'll try to break this down and give you an idea, but the [readme.txt](#) is still the way to go.

.nif & .kif - These are meshes, and they always go within the [Meshes](#) subfolder.
 .dds & .bmp - These are textures, they go within the [Textures](#) subfolder.
 .tga - If used as an icon in the game, put it in the [Icon](#) subfolder. If used as a birthsign, put it in the [Birthsign](#) subfolder. If used as a texture (rarely), put it in the [Textures](#) subfolder.
 .mp3 - If used for music, put it in the [Music](#) subfolder.
 .wav - If used for a sound, put it in the [Sound](#) subfolder.
 .esp - Put these in the [Data Files](#) folder.
[readme.txt](#) - Don't delete these! Keep them, you might need to refer to them again (and again and again and...). I keep mine all within the [Data Files](#) folder. Make sure the name isn't just 'readme.txt', rename it to include the name of the plug-in so you know which one it is for.

There is an added hitch; the modder may have put his files in a separate subfolder within one of the above mentioned folders, just so their files are easy to find. You can't always tell this by looking at the contents of a compressed file. This is why I recommend you extract to a temporary file, at a minimum, before you install a plug-in. An easy way to install a plug-in is to extract to a temp folder. Once extracted look and see what came out. If you find some folders labeled [Meshes](#), [Textures](#), [Icons](#), [Sounds](#), etc, you're in luck. Drag and drop these folders into the [Data Files](#) folder, click 'yes' (it should be ok to overwrite), and that's it. Very easy.

But what if you extract and all you get is a slew of different file types, not even a folder to hold them. Most often seen as a modders resource, in which case you need to figure it out. If you can figure out where they go, go for it. If not, it probably won't hurt anything if you put something in the wrong place, but the plug-in probably won't work correctly (or maybe not at all). If you are not familiar with these file types, and are not entirely sure what each file extracted is exactly for and where it goes, don't mess with it. Ask someone, such as the author of the plug-

in, or post on a forum for help. I know how much fun it is to uninstall and/or reinstall a game or backed up file, but I really don't recommend it.

How do I play this Mod?

When Morrowind starts up and you're staring at the screen there is a button that says **Data Files**, click it. Just place a check mark next to any mods you want to play. Make sure to also check Tribunal and/or Bloodmoon if they're necessary to play one of the mods you load. Once done click 'Ok' and click on 'Play'. Enjoy!

Plug-in conflicts. If two plug-ins are loaded that conflict you will have problems (and you might not know it till somewhere along the story line). There is a utility to check for conflicts, while not perfect, it is very, very, good (see [Chapter 6](#)). Something that usually doesn't provide conflicts but can still cause problems are [Leveled Lists](#). Many plug-ins make use of leveled lists, but only those leveled lists in the last plug-in loaded are used by the game! Leveled lists from other plug-ins are ignored or overwritten! While this might not render a loaded mod unplayable, it generally will lower or limit the game play value. To fix this you can use a utility that merges these leveled lists and allows you to load this singular master leveled list as a plug-in. Very useful. Check [Chapter 6](#) for this also.

Where's the Console?!

To open the console all you have to do is hit the "~", or 'tilde' key. On most keyboards it is the only key to the left of your '1' key. If not there, just look for it. On non-English keyboards you should be able to use the key to the left of the '1' key, regardless of what it is. Now then, how do you use it. Functions are listed below with a brief explanation. A couple things to know though when using the console; you can open the console regardless of whether you have opened up your menus or not (i.e. right-clicked). Right-clicking with the console open, but the menus closed will resume game play, and the console will remain open until you hit the 'tilde' key again. Mess with it a little bit to see how it works. This can be useful; if the console is open and you resume play you will notice that when you 'look' or center the screen upon any object, that object's ID is displayed in the game (not in the console, but in the game). This can be invaluable.

I have compiled a list of functions with a brief explanation below. None of these functions are available for use within scripts. Many, many more commands and functions can be used within the console but are not listed here (see [Appendix B](#) for the full list). As good as this list is or is not, the definitive guide for commands and functions is to be found in GhanBuriGhan's 'Scripting for Dummies', or check out The Unofficial Edler Scrolls Pages website. I highly recommend reading these if you even think about using the console, for any reason, whether you plan on doing any modding and scripting, or not.

Console Functions

This is not a comprehensive list of all functions within the game. Again, you will find that listing within [Appendix B](#). This is a list of functions that can only be executed within the console window (and a couple from the World Menu). They are to be typed in exactly as shown.

***Note:** I left quite a few functions and commands out of this list. Mostly they appeared to be functions that were used by Bethesda for testing purposes but have little to no value for most anyone else. If you must know what they are, then check out Dave Humphries Unofficial*

Elder Scrolls Page (see [Appendix C](#)). He lists pretty much everything so you can really screw things up good with your game if you choose to do so.

Keep in mind: If you use a [Toggle](#) function, that function stays in effect until you change it again (except a few). This **includes** if you start a different game or exit from Morrowind. I don't recommend using Toggle functions and leaving it on/off. Change it back before you exit a game so you don't forget what you did.

Functions

The short hand form of a function is listed in parenthesis (), and may be typed in lieu of the full function name.

BetaComment (BC) - This command can be used to store comments pertaining to an object. Open your `Morrowind.ini` file, find the line `BetaComment=.` Change this to `BetaComment=BetaComment.txt`. Now when playing you can open the console, click on any object, type 'BC' in the console, then add your comment in quotes. The function will output what you type to the `BetaComment.txt` file for later review. When you open the file you will see the comments in the following format: Date, Time, Object's Parent File, UserName, Cell ID, xyz-grid, "Then your comments in quotes". This is a truly useful function for modders.

CreateMaps - This command creates a map of all exterior cells within the game. This can take a long time, make plans for the day, or go to sleep, or come back later. To use this command you must do the following: **1)** Open up your `Morrowind` folder, create a new subfolder titled `Maps`. **2)** Make a back up copy of your `Morrowind.ini` file. Now open your `Morrowind.ini` file, find the entry that reads `CreateMapsEnable = 0`, change the 0 to a 2 (1 is for Xbox). Save and exit the file. **3)** Start your game, open up the console, and type in **createmaps, [yourmod].esp**. Presumably the game will now load every cell one at a time and make a copy. When it's done, to see each map, open up the `Maps` subfolder you created. You will find a 256 x 256 high color `.bmp` map within the folder for every cell.

***Note:** I have seen several statements that this takes longer if you load a game you have played, as the game must load a cell you have been to, then adjust for all the changes to that cell due to your visiting it (I find this odd). It was suggested it is faster if you start a new game, pick your name, and click "ok" to the first message box you get from the guard. Don't follow him, just open the console, type in `createmaps`, and wait. The game will process cells faster because you have not yet visited any cells. I myself received a message saying it couldn't fulfill the function, yet the maps were created. It took little time. I don't know why.*

FillJournal - This will fill your journal with all entries. Be prepared, as it will take quite a while.

FillMap - This will show all the towns on your map.

Help - This will list console commands and also their short hand form.

Output Reference Info (ORI) - Click on an object with the console open, and then use this command to get reference info about it, to include the cell it's in and the .esm or .esp file it's from. Very, very useful.

ResetActors (RA) - This command will reset actors to their starting position coordinates and set their AI package to their default.

ShowTargets (ST) - This will show a selected NPC's target group members

ShowVars (SV) - If no object is selected, this will show all variables and their numerical values, both global and local, in the console. If you center on an object (select it), this command will only show the local variables and numerical values for that object.

StopCellTest (SCT) - This will stop a cell test started with the command TestCells. If you can't get the game to respond to a right click to stop the test, keep trying. It seems the game will pause momentarily every so many cell tested, you can then r-click, and type this in the console. When you stop the test, your PC will remain right where he is, centered in the current cell (this might not be good...).

TestCells - This will start a test where the game will place your PC in the center of each cell to test if the presence of your PC will cause problems with anything in that cell (i.e. scripts, models, etc). It does not test the whole cell for problems, only if your PC presence in the cell causes conflicts. Each cell is tested instantly, so you have little to no control over your PC. I never encountered a problem, so have no clue how the game would respond if an error were encountered. I ended the test before complete. While each cell test is quick, the game goes through every single cell, and this can take quite a while.

***Note:** This takes a while as the game places your PC in every cell and tests that cell. Plus, somehow during this process my character took damage when I tested it (I think I was centered in some lava, but not sure. The process goes quite quick).*

TestInteriorCells - This will test interior cells just as TestCells will test exterior cells. As above, the game is still running while testing, my character died twice. I clicked 'Yes' to reload the last saved game. It loaded, and then continued the test from where I had died. Also, an NPC twice forced a greeting bringing up the dialog menu. The test paused after several cells until I hit cancel. I finally hit 'ctrl+alt+dlt' to end the game.

TestModels (T3D) - This tests all 3D models in the game. Not nearly as long a process as those above.

ToggleAI (TAI) - Turn on or off NPCs AI packages.

ToggleBorders (TB) - Turn on or off ability to see the border between exterior cells. They appear as broken green lines.

ToggleCollision (TCL) - Turn on or off the ability to collide with objects such as walls, rocks, etc (i.e. you can walk through objects). NPCs will sometimes drop through floors too.

ToggleCollisionBoxes (TCB) - Turn on or off colored boxes around every object showing the limits of collision.

ToggleCollisionGrid (TCG) - Turn on or off a mesh like grid over everything showing where objects collide with the landscape.

ToggleCombatStats (TCS) - Turn on or off the display of numerical values as they process during combat (i.e. chance to hit, damage, blocked, etc) in the console window.

ToggleDebugText (TDT) - Turn on or off numbers showing wind speed, timer, and numbers related to movement, etc. I do not know exactly what all are for or what their specific numerical values mean.

ToggleDialogueStats (TDS) - Turn on or off

ToggleFogOfWar (TFOW) - Turn on or off the ability to see areas on you haven't explored.

ToggleFullHelp (TFH) – Displays the ownership and scripts of object you center on.

ToggleGodMode (TGM) - Turn on or off invulnerability. You lose neither health, magicka, nor fatigue. Creature can attack you, but they never hit or damage. Note this doesn't mean you can't die. A script could still set the PC health to zero, thus die.

ToggleGrid (TG) - I'm not entirely sure what this is supposed to do, but on my computer it flooded the screen full of ones and zeros, I couldn't effectively see anything, and slowed my game to a screeching halt. It took almost 20 minutes for me to bring up the menus so I could toggle it off. Try this at your own risk (and don't say I didn't warn you).

ToggleKillStats (TKS) - If on, the game will keep a running score of what you kill and how many in the console.

ToggleLights (TL) - Toggles on or off light sources.

ToggleLoadFade (TLF) - Turn on or off the fade effect when loading/changing cells.

ToggleMagicStats (TMS) - Turn on or off the display of numerical values as they process during magic and spell processing in the console window.

ToggleMenus (TM) - Turn on or off the ability to bring up any menu.

TogglePathGrid (TPG) - Turn on or off the display of AI Path Grids. These appear as double yellow lines running from node to node. Nodes appear shaped like red gumdrops.

ToggleScriptOutput (TSO) - Not sure what this command does.

ToggleScripts - Turn on or off whether scripts execute.

ToggleSky (TS) - Turn on or off the display of the 'sky'. If off, the sky is black (kinda cool).

ToggleStats (TST) - Turn on or off both combat and magic numerical value display as they process.

ToggleVanityMode (TVM) - Turn on or off the ability to switch to 3rd person view with the 'Tab' key.

ToggleWater (TWA) - This did not appear to do anything in exteriors.

ToggleWireframe (TWF) - Turn on or off the display of everything in wireframe mode (no textures are rendered or displayed).

ToggleWorld (TW) - Turn on or off the display of all landscaping. It is still there, you just can't see it. Everything appears as endless ocean.

Use these commands at your discretion. Many are useful for when trying to debug something you've done in your mod, but doesn't work, or at least not the way you want it to. Some I'm not sure what they are supposed to do, or what it is they are doing. Some of the Toggle commands are useful for testing how the game responds to changes you've made with a mod, others most likely were useful for the developers, but provide little more than confusion or amusement now, and probably should have been, but were not, removed when the game was released.

Lastly, remember that many commands are not listed here that can be used in the console window. You can add/delete objects and spells, modify skills and attributes, resurrect NPCs if killed but shouldn't have been, and in general make many modifications to the game. For those who like to cheat, the console is easy and available. There is also nothing wrong with cheating, to each his own. But if you like to mod, you will find the console invaluable ingame.

Call the exterminator! Bugs everywhere!

Bugs. We all hate them. It's inevitable that they will show up, and the more complex and bigger your mod, the more likely you will have them, and more of them! How do you get rid of them? The simple answer is you have to figure out what is not working right, go back into your plug-in in the CS, figure out what you did wrong, and fix it. No simple way to do it either, you just have to do it. But there are a number of things you can do to simplify this process.

Write it down! First and foremost; write down every single addition, change, or deletion you make to your mod. Sounds like a lot of trouble right? It is, and it will draw the modding process out timewise, believe me. So what, do it anyway. I have caught bugs before I even hit the save button by looking over my notes as I work. Get yourself a notebook, or one with sections. Each page or section can pertain to changes you have made to the world pertaining to an area, or to an aspect of the game (like spells, skills, races, etc). Be meticulous, note everything you do. If you

write down you modified an object, write down its' name **AND** it's **ID**, exactly as it's spelled. Put down where it is in the world; the cell, outside or inside what building/dungeon, on what floor, on top of what, etc. Added a script to it? Write it down. Shifted this thingy around to make room for said object? Write it down. Made a mistake and dirtied something up? Write it down! You will have to clean that out (Dirty mods are explained later on, but make the note! It's important!). Being able to look over your notes and catching problems before you save a mod is a real stress reducer. A lot easier than spending untold amounts of time ingame playing your mod before you come across the bug, and then pulling your hair out trying to figure out what you did wrong. Your notes also give you the basic draft for writing your readme.txt to include with your mod. You will include one, won't you?

Make new objects! Don't make changes to objects you find in the game, make a new object! (refer to [The Objects Window](#)). When you make a change to an object in the CS, regardless of whether it is an object reference in the Render Window, or you click on a parent object in the Objects Window, any change you make to that object affects every single instance of that object everywhere in the game! This can't be stressed enough! If you do this you might just be screwing the game up somehow so it can't be played or completed. Or possibly rendering your mod completely incompatible with almost every single other mod created that's out there. You will see this again, but for now, just take heed.

Don't make a dirty mod! If you click on an object in the game to modify it in some way, then change it back to it's prior state of existence, the CS notes that you changed that object in some way and stores that information in your mod. The CS can not distinguish between an actual change you make to something, or the fact that you changed something back to the way it was, only that an object has been in some way altered. This probably won't screw up your mod, but it might someone else's! It might also add size to your mod, making it bigger. A few of these won't add noticeably to a plug-in's size, but if you are one of those who doesn't care about this, you probably have many such dirty changes. Which means there's a greater chance your mod is both buggy, and much bigger in size than necessary, which means a lot longer time for download (What? You think everyone is running a DSL?). Plus the more likely your mod won't be compatible with other mods out there. People don't like dirty mods. If this does happen, don't worry, it's not hard to fix. There are utilities to help, or you can just use the CS itself to clean those dirty rotten stinkin' entries out of there.

Resist the urge to create uber mods. When you first open the CS it's tempting to create an uber-item so you can waltz through the game, and a lot of people do at some point. I did it (after my first dance through the game), and it was fun. For a while. Then the enjoyment wore off. But some people seem to think their uber-item is so powerful and cool everyone should have it too, and they upload it for the masses, joining the ranks of uber-items for download. Most websites now have enough of these they don't need more. There are also plenty of tweak mods out there. What are these? Mods that change the entire aspect of the game in almost every way to make it harder, or easier. These are really great and take a great deal of work, and lots of knowledge and effort. These are a real challenge to create. But there are plenty, why create what someone else already has? If you can make one in a new and unique way, or different from others, I say go for it! Originality is something the mod community loves to see. But do you want put effort into something, only to upload it and find out someone else did the same thing some time ago? I'm not trying to dissuade anyone from attempting anything. Do what you want and like. It's up to

you. But I like to plumb the depths of unexplored territory, or at least do something in a way no one has. But that's just me.

Use forums on websites to get help. Can't figure out how to do something? Can't figure out why something isn't working the way you want? Want some advice or a mod tested by someone else for insight? Join the forum on a website you like, and post questions, troubles, or search for some advice there. You might find out some things you didn't know, or possibly someone already asked your question, and got an answer (which you need). People on forums are usually, reasonably, friendly, almost always helpful, and you might be helping someone else out too. This is a personal choice, and up to you. I myself gleaned enormous amounts of useful information that was invaluable from forums at several websites, and usually got exactly the help I needed when I posted a question asking for it. You just about can't go wrong.

Can I play all 5-gigazillion mods I installed?

The short answer, yes. The long answer, most likely yes. The game's actual limit is 255 files loaded at one time. That's both .esp and .esm files. I've seen people state they have loaded and are running over 200 plug-ins. It's possible. However, the ability of your system to handle the plug-ins you load, regardless of how many, is the real limitation. Some people create mods that really test your machine's ability to churn out those FPS. A slower machine might have trouble with some mods that add lots of scripts to be executed globally, constantly. Or have too many objects animated and interacting all at the same time on screen. Having a ton of plug-ins loaded and playing at the same time could produce these conditions. The real answer is not how many can you have loaded on your machine, but which ones can you load that your machine can handle together. This is just a matter of trial and error unfortunately, but you will find that you can load up quite a few and run them all as you wish. The only real problem is when plug-ins conflict, as I mentioned earlier.

Chapter 3 Toolbar Description

The Toolbar

This is the toolbar. From here your mod is given life, your imagination will become virtual reality, and your life will be shortened by 1.7 years due to stress. And that is the point of this manual, to reduce or eliminate that 1.7 years of life lost from stress. So let's start with the actual Toolbar and see what we can find to do on it.



The toolbar is not exactly the average windows toolbar, which is quickly obvious if you click on any of the menus to see what drops down. I will discuss each menu and each button in order. Those menu functions that open a window I will go into greater depth later in the manual. Many of the menu listings are also accessible by clicking one of the buttons, and some also have a shortcut key. Let's start with the menus.

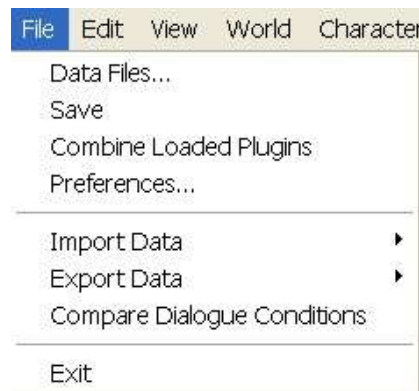
The Menus

File Menu

File->Data Files...

This will open a window that allows you to view all plug-ins (.esp) and master files (.esm) installed on your computer. Master files are listed first, by date (oldest at the top). Then plug-ins will be listed (again, oldest at the top). To load an .esm or .esp simply double click it, and a check mark appears next to it. This means it will get loaded into the CS when you hit the "ok" button at the bottom. Some things to remember:

- When you load a plug-in, any .esm that it is dependent on will also be loaded, regardless of whether or not you marked it for loading. This is automatic and you cannot prevent it (this can be changed with a utility, see [Chapter 6](#)). This is true even when loading multiple plug-ins with different master file dependencies. If you don't mark the proper .esm files to be loaded though, you could get an error message. This happens sometimes with certain .esp files. I don't know why.
- If you load one or more plug-ins, and load one or more master files, when you hit the save button, the active plug-in will become dependent upon **all** master files loaded. You cannot undo this with the CS (hello...utility, see above bullet).



TES3 file list. Pick what plug-in listed you want to load. Once you've made your decision on what to load, you must highlight one file and click the "Set as Active File" button. This plug-in will be considered *active*. Any changes made to anything will be saved in this active plug-in. You can still see data from any other plug-ins that you also load, you just cannot save any data to those plug-ins. Once chosen just click the 'ok' button and the CS will load all marked plug-ins and master files. If you load only master files you do not need to make one active, the CS will load the .esm files and any changes you make will become a new plug-in (you will be prompted to name it if you hit save). If you forget to set a plug-in as active, the CS will bring up a prompt stating this and ask if you wish to continue. If you click "No", the CS will close. If you click "Continue" the CS will generate errors (usually animation errors). Start over.

Created By. Who created the .esp file. You can enter your name, or ID, in this block.

Summary. Here you can write a small summary about your plug-in. If it's small with few changes you probably don't need to say much. You should at a minimum list what this plug-in is about, and a version number.

Parent Masters. Here you will find listed all master files that a plug-in is dependent upon. Note that master files can be dependent too.

Details. This button brings up a window that allows you to see all the changes you've made to your mod. The window lists all the IDs of changed data within a plug-in. If you see something here you don't remember creating or doesn't have any effect on your mod, your mod is dirty and needs to be cleaned. Refer to [Cleaning the Unclean](#) for more information.

File->Save

Click and save your plug-in. I have noticed that when working within some windows (Dialog, Scripting, and Cell Path Grid, etc) if you click the Save button, then close the window, the changes you make are saved only within that aspect of the game. You must still click the save button to save it to your plug-in once that window is closed. An asterisk will be present next to the Name of the plug-in at the top of the CS window if changes have been made but not yet saved.

File->Combine Loaded Plug-ins

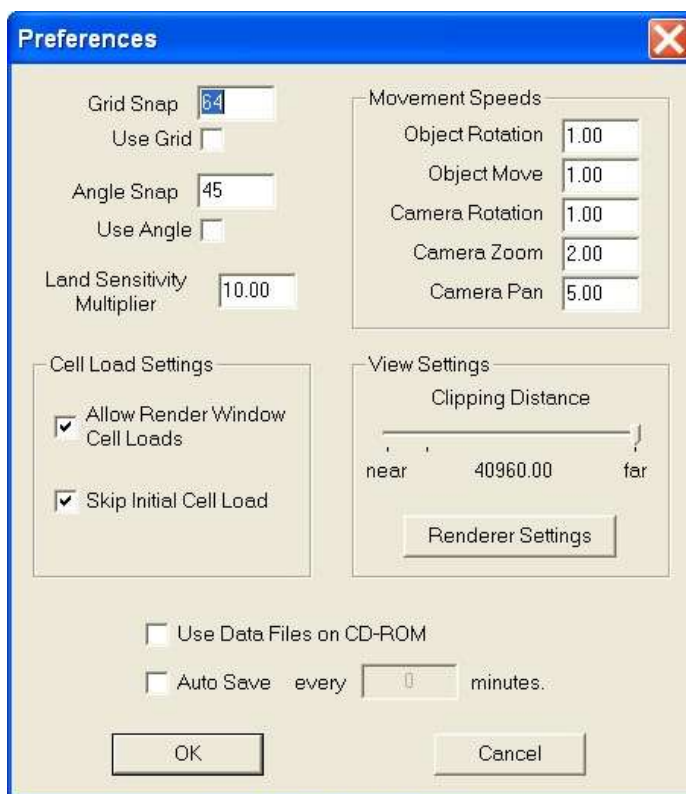
Load any and all plug-ins you want combined. Click on this option under the File menu. A prompt will appear asking if this is what you want to do. Click yes, and a window will appear. You will have to provide a name for the new combined plug-in that will be created. Do so and hit Save. Your new combined plug-in will become the loaded plug-in, displacing all others, become the active file, and will appear at the bottom of the list in the Data Files window. Consider this option carefully before you do this, combining plug-ins that conflict will produce problems.

File->Preferences

This will bring up a window with options to change the rate at which you can move objects within the Render window, move your view within the object window, load cells as you work, and what you can view of the exterior/interiors.

Grid Snap/Use Grid. When the Use Grid box is checked, objects moved by you in the Render Window will move a number of units determined by the number entered in the Grid Snap box. The CS states to use multiples of 8 (8, 16, 24, 32, etc), though any number can be used. This can be of use in moving objects quickly and getting them to line up with other objects easier.

***Note:** This is useful when placing large static objects, like when putting together a building (interior or exterior). But for small objects it's probably easier to do manually without this enabled. Really it's a matter of choice, what you feel works best for you.*



Angle Snap/Use Angle. When the Use Angle box is checked, objects in the Render Window will rotate by you a number of degrees determined by the number entered in the Use Angle Snap box. Remember there are 360 degrees in a circle. See above *Note* about use.

Cells Load Settings

Allow Render Window Cell Loads. When unchecked, the CS will only load cells, and their objects, that are visible in the Render Window. Cells not fully visible will not be loaded; wilderness cells will not be created outside of the Render Window. If checked, the CS will load visible cells (even partially visible ones), and a two cell 'buffer zone' around each and every fully, or partially, visible cell in the Render Window. This may slow down some slower systems.

Land Sensitivity Multiplier. This is the rate at which the landscape mesh/wireframe is changed, or moves, in response to your mouse when landscaping. The default is quite slow, good for small refined changes. Increase if you wish to make large changes quicker.

Skip Initial Cell Load. When a plug-in is opened, cell data is loaded for cell 0, 0. Checking this prevents that from happening. I have found unchecking this to have a minimal effect in load time for a plug-in, but it may help on slower systems.

Movement Speeds. These are the speeds at which objects or the viewpoint moves or rotates in the Render Window.

View Settings

Clipping Distance. In simple terms, effectively, this is how far you can view into the distance. Move the slider bar closer to near for slower systems for better and/or quicker response from the CS. There is a single tick mark approximately $\frac{1}{4}$ over from the left on the bar. This is exactly 8192 units in viewable distance (the length of a side of a cell).

***Note:** I like have it set close (for a single cell) when adding/deleting or modifying objects in a cell. Then about halfway when adding buildings and structures for a town or village that spans more than one cell. When landscaping I set it to max and use wireframe mode. I change this slider quite often as I work. Use what works for you.*

Renderer Settings. Don't even mess with this. Even clicking on the button will take a while on a quicker system. This allows you to use Hardware or Software to render 3D images, and also to change the drivers used, and the screen resolution used by the CS. Unless you fully understand what this does, leave it alone. I mean it.

Use Data Files on CD-ROM. When this box is checked and you start the CS, the CS will look for the Construction Set CD in your CD tray, and will look for files on it when you click buttons to assign mesh, icon, texture, etc. files to objects. If the CD is not found it will let you know and instead always look by default in your `Data Files` subfolder on your hard drive. (see [Copying Files from CD to HD](#))

Auto-Save. Check this box if you prefer the CS to auto-save your plug-ins, and enter how often in the minutes box. I have this feature off to prevent saving something stupid I didn't realize I did making my mod dirty or unplayable. Do what works for you.

File->Import/Export Data

You can use these functions to move object data between mods. Be aware that not all data for many objects is carried over during import/export.

File->Compare Dialog Functions. Clicking this brings up a window where you can click an `.esm` to compare dialog against. Presumably to check for conflicts. I did not test this to any great degree.

File->Exit. Quit the CS.

Edit Menu

Edit->Undo/Redo.

Undo a mistake you made, or redo something you clicked undo on, but decide to keep.

Edit->Cut/Copy/Paste Render.

This is your standard cut, copy, and paste. It only applies to objects in the Render Window.

***Note:** You can use the mouse pointer to box in multiple objects, thus selecting them as a 'group' and treating them as such for cut/copy/paste. Control + c, x, or v keys will work for both objects and text.*

Edit->Paste in Place.

This will place cut/copied objects at the selected spot, upon the ground or floor.

Edit->Duplicate.

Clicking this will create a duplicate of selected object(s). The duplicate will appear exactly on top of the object(s) duplicated.

Edit->Find.

Brings up the Find window. This window has a drop down menu listing of all objects in the plug-in. You choose which one you wish to find and it searches references until it finds one. It then loads the object's first reference found in the Render Window in the cell it found it in. You can use the next set of menu options to see the next or previous 'found' reference. If an object has no references in the game, it will do nothing.

Edit->Find Next/Previous.

This will cycle through the objects references 'found' with the Find option.

Edit->Find Text.

This option brings up a window with a search box, a button to the right of it labeled 'Find text' to start the search, and three tabs. These tabs are labeled Dialog, Script, and Object. It will list all dialog, script, and objects in the appropriate tabbed window according to your search criteria (what you typed). The function appears to accept results that may not be clearly related to your search criteria (though usually are).

Edit->Search & Replace.

Edit	View	World	Character	Game
Undo			Ctrl-Z	
Redo			Ctrl-Y	
Cut Render			Ctrl-X	
Copy Render			Ctrl-C	
Paste Render			Ctrl-V	
Paste in Place			Ctrl-Shift-V	
Duplicate			Ctrl-D	
Find...			Ctrl-F	
Find Next			F3	
Find Prev			F2	
Find Text				
Search & Replace...				

This brings up a small window with two drop down boxes. The first is the object you wish the CS to find all references for, and the second is the object you want the first to be replaced with. There are two check boxes. One will execute the function in the current cell, the other will only operate on a selected object (which should be listed in the first drop box) or group of objects.

View Menu

View->Toolbar.

This will display or not display the buttons beneath the menus.

View->Statusbar.

This will display or not display the bar at the very bottom of the CS window displaying information on currently selected objects and actions.

View->Render Window.

This will display or not display the Render Window.

View->Object Window.

This will display or not display the Object Window.

View->Cell View Window.

This will display or not display the Cell View Window.

View->Current Cell Only.

Completely grayed out. Most likely a feature turned off, but was of use to the developers (but not us apparently).

View->Markers.

This will display or not display North, Door, Temple, Divine, and Travel markers in the Render Window.

View->Collision.

This will display or not display the collision grid in the Render Window, showing where the PC will collide with objects. This is only for objects, not landscape.

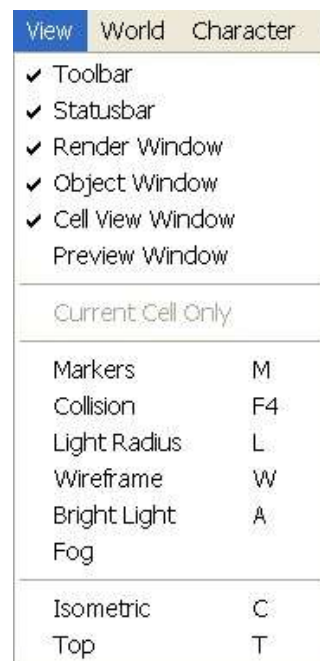
View->Light Radius.

This will display or not display the radius of light emitted by an object by the presence of a tri-colored globe encircling an object that emits light.

View->Wireframe.

This will display or not display the landscape and all objects in wireframe mode.

View->Brightness.



This will brighten everything within the Render Window. By default the Render Window is somewhat dark. To better see objects within it, click this option. Just keep in mind it will look different when ingame; if you forget to put any lights in because you 'brightened' your Render Window...

View->Fog.

This will display Fog as a weather effect in the Render Window so you can see what it might look like in that cell.

View->Isometric.

This will center your viewpoint upon a selected object at ground level, facing north

View->Top.

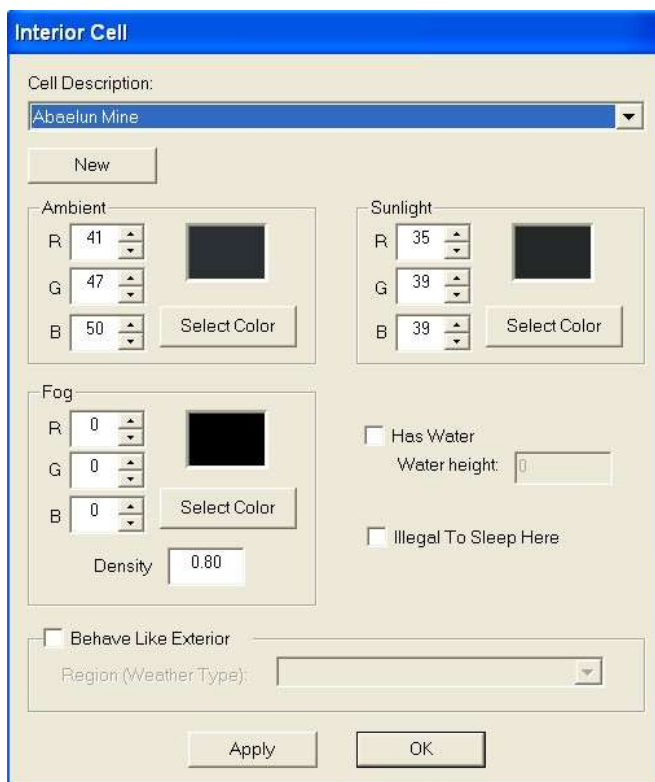
This will center your viewpoint upon a selected object above said object, looking straight down at it (north is up on your screen).

World Menu



World->Interior Cell.

This brings up a window to create an interior cell. It does not create anything within the cell, it only allows one to adjust the lighting, allow sleep, set a water level, or, believe it or not, create an interior cell and set it to act like an exterior cell (gasp!). Yup, just like in Tribunal. A drop down menu at the top allows you to choose an interior cell to make adjustments too, or click on 'New', which brings up a small window to name your new cell. Click 'ok' and the new cell name will now appear in the drop down menu. You can now set the color of ambient light, adjust the amount of sunlight (actually, this is more along the lines of what you might consider 'normal' light, though you can set it to zero), and the amount of Fog if any. If you want water in your cell, check the box for water and enter a number in the box next to it. This number will be the grid coordinate level on the z-plane that the water level is set to. If

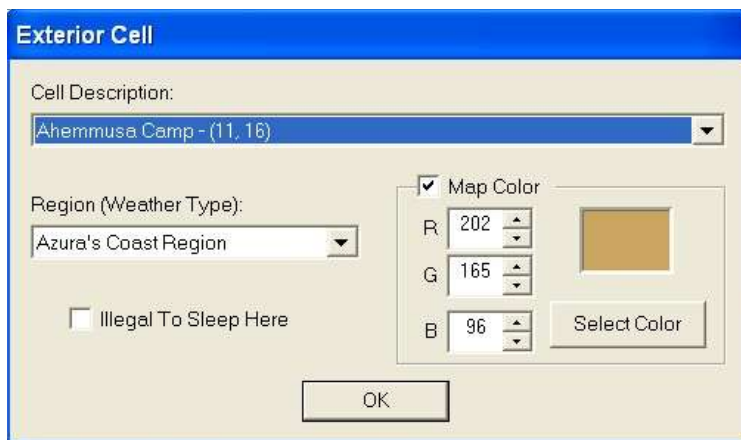


unchecked, there will be no water in this interior cell. Also, check this before you build your cell. I have seen several statements that allude to certain difficulties when trying to add water to a cell already built. This seems to be the case, as I could not get water to appear in a built cell. Lastly, if you don't want the PC to sleep in this cell, check that box. This window creates the existence of the cell, and sets parameters for it, but does not *build* it. When done, go to the Cell View Window and find your new interior cell, d-click it. It will load in the Render Window, but will be blank (all gray). You can now build the physical portion of your cell. Remember, you use this function to create the characteristics of the cell first, and then you can actually build it with objects in the Render Window.

Note: You must have Tribunal or Bloodmoon to be able to set an interior cell to function as an exterior cell.

World->Exterior Cell.

This will allow you to set the characteristics pertaining to an exterior cell. At the top is a drop down menu showing what exterior cell you are modifying. Below that is a drop down menu to set the weather based upon the region you assign with it. The Map Color box, if checked, allows you to set a color on the game map representing an exterior cell. This is usually only done for town cells. Below that is the Illegal to Sleep Here check box. Don't confuse this window's functions with World->Regions. You use the Regions to set the weather for a region. You use this window to set a region you want to use the weather for in this cell (regardless of whether the cell is assigned to that region). Confused yet? Don't worry, you'll figure it out and it won't seem so complicated.



World->Regions...

This allows you to change the settings for regions, including weather, being attacked while sleeping, and sounds. A region is a collection of cells that are of similar make (ie. the Grazelands, Ascadian Isles, etc), but they don't *have* to be similar. A town is part of a region, and a single cell, or several cells, could be quite different yet still be considered part of a region.

Region ID. This drop down menu shows a list of all regions in the game.

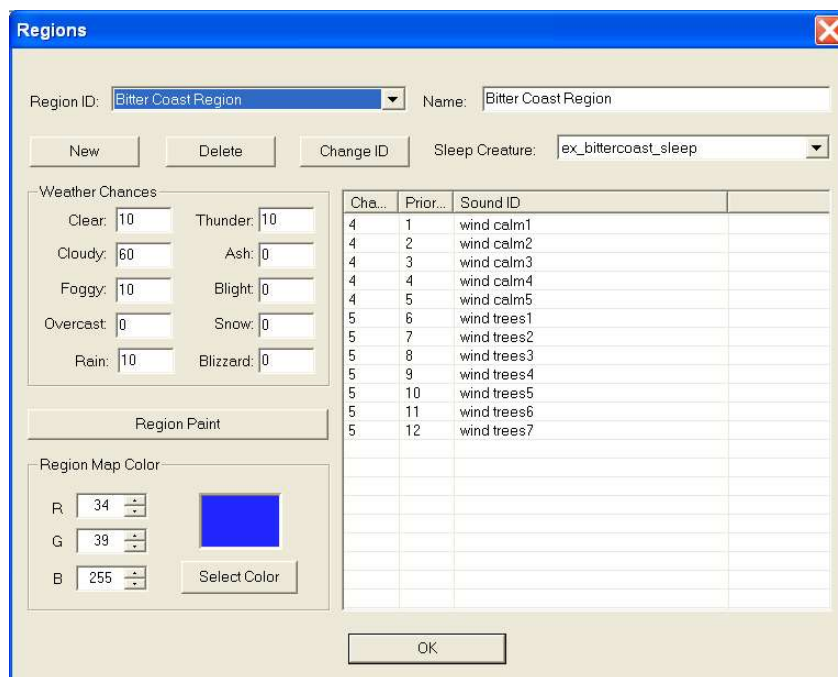
Name. This is the Name of a region as shown in the game. Usually the same as the ID.

New button. Click this to create a new region.

Delete button. This will delete the region shown in the Name ID drop menu.

Change ID. This will change the Name ID. Easier than deleting an ID and creating a new one (Bethesda should have included this button on other object windows too).

Weather Chances. This lists all the different weather types for a region. Each has a percentage associated with it. This is the percent chance of that weather type taking place in that



region. These percentages MUST add up to 100%. If you set a weather type to 0%, it will never appear.

Sleep Creature. This is the creature list used for when the PC is awoken during sleep by an attacking creature. This can be specific creatures, or a leveled list.

Region Paint. Click this to bring up the separate Region Painter Window. Here you can 'paint' each cell in the world a color assigned to a region. Highlight a region listed in the window to the right and any cell you touch becomes the same color as that assigned to that region. This tells the game what region a cell belongs to.

Region Map Color. This is the color palette you can use to assign a color to a Region. Assign a color to a region before you open this window to assign cells.

Sound. The large box shows sounds that play ingame for this region. The first column is the percent chance of that sound playing. The second column is the priority to play that sound if more than one sound meets the percentage chance. The lower number plays. The third column is the name ID of a sound. To add a sound to this list open the Sound Window, then just drag and drop it into the list. It will be placed at the lowest priority. To delete a sound, highlight it, and push the 'delete' key. There is no way to adjust priority once a sound is added (what was Bethesda thinking?).

World->Go To Cell.

Brings up box of cell listings. Choose a cell and click 'ok'. This loads a cell in the Render Window as if you had d-clicked it in the Cell View Window,

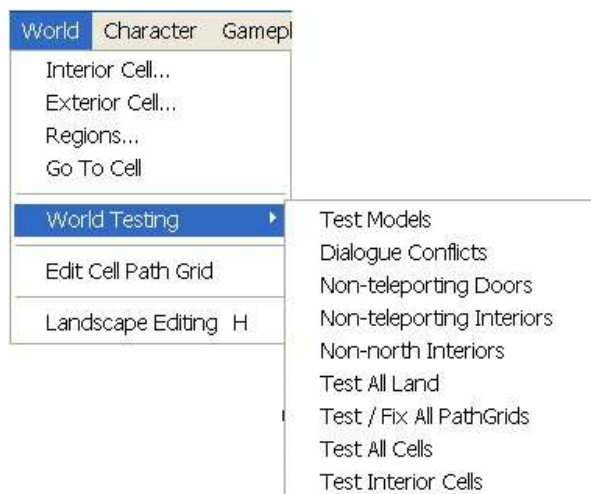
World->World Testing.

This brings up a further menu enabling you to test certain aspects of the plug-in world data for errors and conflicts. Some things to keep in mind, each of these tests will test the entire game world and your plug-in. A couple of these will take some time. If you just made a small change, wait and test your plug-in ingame, or wait till you have more changes to check and then run a test. These are great functions to help you debug your plug-in.

Test Models. This will test all 3D models of all references in the world and your plug-in. If an object is missing part of it's 3D mesh or texture, this should discover that and when done testing will provide you with a list of all such errors it found.

Dialog Conflicts. This test will check all dialogs responses and check for syntax errors. If a response causes a conflict somehow this will return a list of the actors that can give the response. You can then go and fix it instead of discovering this fact during game play.

Non-Teleporting Doors. This will test and return a list of all doors placed in exterior cells that are not linked (marked to teleport). This lets you know you forgot to link a door to another cell (or place).



Non-Teleporting Interiors. This will test interior cells to see if they are linked to another cell via doors marked to teleport. This does not test to see if a cell is linked to an exterior cell, only that it is linked to another cell, interior or exterior. Thus you could have a large dungeon split into 4 areas, all linked together and forget to add doors to get in or out of the dungeon. This function would probably miss that fact as it would find you had linked the different areas (cells), but doesn't test for where.

Non-North Interiors. This test will check to see if any interior cells do not have a northmarker, and will return a list of those you that don't.

Test All Land. This will return a list of any cells that have landscape problems. It will not tell you what the problem is, only that one exists. You may have to look carefully for it.

Test/Fix All Path Grids. This will test all path grids. Any nodes that do not have a path connecting to it will be removed.

Test All Cells. This will test every cell in the game world, landscape and objects, checking for any possible errors. This can take a *long* time to finish.

Test Interior Cells. This functions in the same way as does Test All Cells, except it only applies to interior cells. Takes quite a while.

World->Edit Cell Path Grid.

Refer to [The Edit Cell Path Grid Window](#). This brings up a box you can use to add nodes to the landscape, and link them. This creates paths for NPCs and creatures to follow.

World->Landscape Editing.

Refer to [The Landscape Edit Window](#). This is what you use to modify and change the landscape in the CS.

Character Menu

Charater->Race.

This brings up a window within which you can edit, create, or delete a race. This does not create, edit, or delete a body for a race, this only defines the data about a race. Refer to [The Race Window](#) for information on modifying or creating a race.

Charater->Skills.

This brings up a window to modify skills, somewhat. You cannot add or delete skills. Nor can you change what actions are associated with a skill. Refer to [The Skill Window](#) for more information.

Character->Class.

Refer to [The Class Window](#) for more information.

Charater->Birthsigns.

Refer to [The Birthsigns Window](#) for more information. The Birthsigns Window allows you to assign spells, an image, and a description to a Birthsign for someone to choose from when



creating a new PC.

Character->Animations.

Here you can review the different types of animation used to animate actors. I have no knowledge of dealing with animations, so will not attempt to confuse you or myself with a dissertation about it.

Character->Factions.

Refer to [The Factions Window](#) for more information. You can open more than one Faction Window to compare different factions.

Character->Dialog.

Refer to [The Dialog Window](#) for more information. You can open more than one Dialog Window to compare different dialog strings.

Gameplay Menu

Gameplay->Magic Effects.

The Magic Effects Window allows you to change how magic appears and sounds ingame when cast as a spell, the particle visual effect, and the icon showing it as active. This window does NOT change or alter spells! It changes the ingame visual effects associated with a 'spell effect' that is used to create spells. Refer to [The Magic Effects Window](#) for more information.

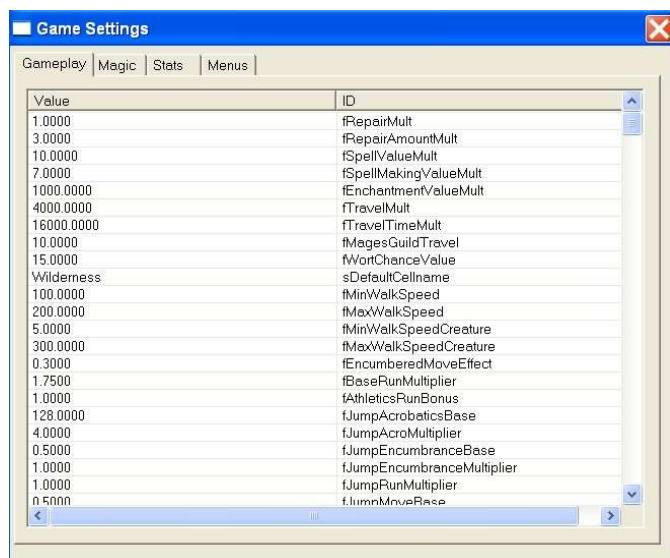


Gameplay->Settings.

There are four tabs on this window; Gameplay, Magic, Stats, and Menus. Each tab displays similar game settings. I will not describe here what each setting does or how it relates to the game. A comprehensive list of settings and descriptions appear in [Appendix B](#).

Gameplay->Scripts.

Refer to [The Scripting Window](#) for more information. You can open more than one Scripting Window to compare different scripts.



Gameplay->Edit Start Scripts.

You can add or delete a script from this menu. Any script on this menu is begun as a global script from the start of the game. *It is highly recommended you do not make any changes to any scripts on this list, nor delete them. Think carefully before you add a script to this list.* If you

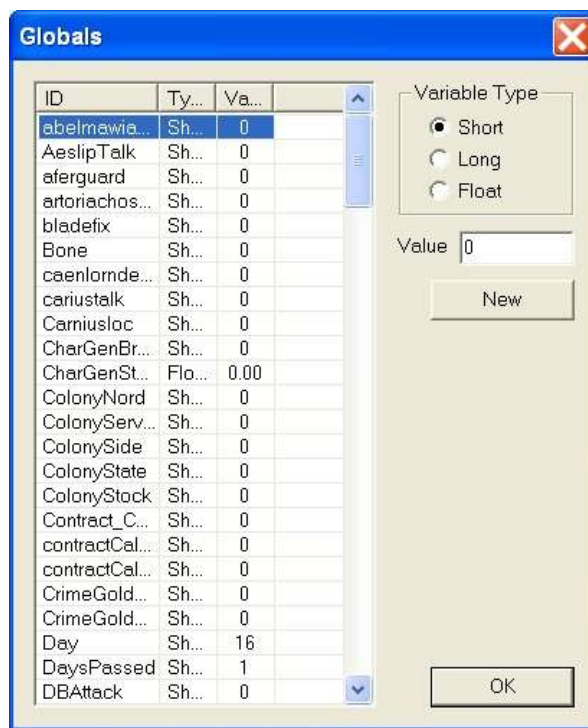
[Table of Contents](#)

want a script to run as a global script, my advice is to place a starting script on a reference somewhere the PC is guaranteed to go in the game, thus ensuring it will execute (see [Scripting](#)).

Gameplay->Globals.

This is a list of global variables in the game that can be referred to in scripting. These variables are different from a variable declared in a global script in that the game constantly tracks these variables. A variable declared in a global script is only tracked while the script runs. To add a new global variable simply click the new button, input the name, and choose the type (Short, Long, or Float). If you wish for the variable to start with a value other than 0, input that also.

***Note:** The game has a ton of global variables that it is already keeping track of (as you can see). This contributes to your FPS, or lack thereof. I don't recommend creating and using global variables unless necessary.*



Gameplay->Sounds.

Refer to [The Sound Window](#) for more information. The Sound Window allows you to assign an audio file (.wav) as a sound. The game will not recognize a sound till you assign it here as such.

Gameplay->Sound Gen...

Refer to [The Sound Generator Window](#) for more information. This window allows you to assign sounds to creatures. Note the sound must exist already in the Sound Window (directly above).

Help Menu

There are only two options, **About** tells you the version of the CS and gives you a picture (seen as the first page of this manual). There are two versions: if you have vanilla Morrowind, you have v1.0, with Tribunal or Bloodmoon then it is v1.30. The other is the CS **Help** file. What can I say...the help file has its uses (which are many...), but is somewhat poorly presented. It is of limited use if you don't know what it all means. Some of the information is just plain wrong even. At least it has several tutorials. For a newbie, it can be daunting just trying to understand and interpret it all.

The Buttons

[Table of Contents](#)

[Table of Contents](#)

The buttons are merely shortcuts to commonly used functions within the CS. I will not go into depth for each function associated with one of the buttons, I will only specify what it does, or refer you to the section that describes that function.

Data Files



Opens the Data Files Window. Refer to [Files->Data Files](#).

Save



Saves your plug-in. Refer to [Files->Save](#).

Preferences



Opens the Preferences Window. Refer to [Files->Preferences](#).

Undo



Made a mistake? Just click this button.

Redo



Oops, you meant to do that, it wasn't actually a mistake. Redo what you just undid.

Use Grid Snap



Toggles on the Grid Snap function. Refer to [Files->Preferences](#).

Use Angle Snap



Toggles on the Angle Snap function. Refer to [Files->Preferences](#).

Landscape Editing



Click this button to bring up the Landscape Editing engine. Be advised, this will prevent most functions in the CS except those related to landscape editing, as this engine is a system resource hog. Refer to [The Landscape Edit Window](#).

Edit Cell Path Grid



Click this button to bring up the Edit Cell Path Grid Window. This will allow you to generate a default set of nodes and paths, or place them manually. Refer to [Edit Cell Path Grid Window](#).

Brightness



Toggles brightness in the Render Window. Refer to [View->Brightness](#).

Fog



Toggles the Fog weather effect in the Render Window. Refer to [View->Fog](#).

Dialog



There are many functions and different drop down menus within the Dialog Window. There are also three tabs. I will not explain how to do dialog in this section. Refer to [The Dialog Window](#).

Scripting



Again, I will not go into depth on the Script Window, but will only explain the buttons and menus on it. To see how to write scripts refer to [The Scripting Window](#).

Script Menu. Here you can open a script, create a new one, or delete a script.

New. Opens a blank script in the script window.

Open. Opens a Script Pick window showing all scripts in alphabetical order.

Previous Script. Open the previous script in the Script Pick window.

Next Script. Open the next script in the Script Pick window.

Save. This will compile the script you're working on, effectively saving it as a script. You must still click Save on the CS toolbar to save it within your mod.

Recompile All. Clicking on this will prompt the CS to recompile all scripts in the Script Pick window, to include any that are open. **This will cause problems**, possibly screw up a mod (Odd that it is even included). Regardless, I don't recommend *ever* using this button.

***Note:** If you do this, your mod can be fixed. All the recompiled scripts will be listed as part of your mod in the Details listing. You can remove these problem scripts and restore your mod to workable status (see [Cleaning the Unclean](#)).*

Delete. Click this and the Script Pick window opens. D-click on a script and a "D" will appear next to its name. This does not delete an open script unless you choose it from the Script Pick window. If you do the script window will turn gray.

Edit Menu. Here you can click on Undo or Redo. You should know what these do.

Help Menu. Bring up the Help Window, but only those parts that pertain to scripting.

Open button. As Open above.

Compile Script button. As Save above.

Previous Script button. As Previous Script above.

Next Script button. As Next Script above.

Recompile All Scripts button. As Recompile All above.

Delete Script button. As Delete above.

Down Arrow button. Clicking this exits the Script Window.

Sound



Used to assign sounds so the game will recognize them. Until you load a sound for use in the game with this window, the CS will not recognize them. Once that is done, the sound can be

used. This does not assign sounds to object/actors, that is done with the [Sound Generator Window](#). I will not go into depth on the Sound, please refer to [The Sound Window](#).

Chapter 4 Windows

In this chapter I will discuss the three main windows the CS uses to function (i.e. Object, Render, Cell View), plus I will go into depth on quite a few others that in the last chapter I only provided limited information on what you could do in the window, not how to use the window to accomplish something (i.e. Dialog, Scripting, Sounds, etc.).

The Object Window

Most objects in the game that can be considered an 'item' are listed within the Objects Window, grouped under a tab, though many are not (Path nodes, scripts, Birthsigns, etc can be

considered objects in some ways, and are things that have to be created, but most do not have a physical representation in the game world). Some items you can interact with, others you can't.

Object Window									
Light	Lockpick	Misc Item	Probe	Repair Item	Static	Weapon	NPC	Creature	
Leveled Creature		Spellmaking		Enchanting		Alchemy		Leveled Item	
Activator	Apparatus	Armor	Body Part	Book	Clothing	Container	Door	Ingredient	
ID	Count	Name	Script	Model	Persis...	Block...			
i\act_black_load01	0	Dark Crevice		i\In_Lava_Bl...	no	no			
i\act_black_load02	0	Dark Crevice		i\In_Lava_Bl...	no	no			
i\act_black_load03	0	Dark Crevice	passageScript	i\In_Lava_Bl...	no	no			
i\act_black_load04	0	Dark Crevice		i\In_Lava_Bl...	no	no			
i\act_BM_axe_hrothmu...	1	Hrothmund's ...	HrothmundAxe	w\W_Nord_...	yes	no			
i\act_BM_beast_parts	1			f\act_BM_be...	yes	no			
i\act_BM_earth_parts	1			f\act_BM_ear...	yes	no			

Object Conventions. There are certain conventions to keep in mind when working with Objects in this window. Some of these apply when working with other aspects of the game too, usually in how it is named.

- **References.** When an object is created it is not placed in the game, but it does exist. When you place an item in the game by dragging it into the Render Window, you place a *reference* of that object into the game world. Think of it as a clone. All *references* of an object are exactly the same, and *some* changes made to one *reference* will change *all other references in the game, including the original object!* This may be the most important thing you can remember when working with any object! If you need to make use of an object already created, but need to change ANYTHING about that item, *make a new Object!* This will prevent you from changing something in the game that breaks a quest or changes how the game operates, or changes your mod in some way that prevents anyone from playing your mod loaded with other mods. You *can* change a few details of a reference, but only a few (only that data below the Reference Bar).

Note: Each reference of an object actually does have a specific separate ID assigned by the game so it can differentiate it from others of the same object, with an 8 digit number added to the Name ID. Thus the first reference of *flora_tree_01* would be *flora_tree000000001*, the second reference created would be *flora_tree0100000002*, etc. The CS does not display this additional number though, you can find it by opening the console ingame, and clicking on any object. The full reference name will be displayed at the top along with the extra identifying number.

[Table of Contents](#)

[Table of Contents](#)

- **Creating New Objects.** Don't create a new object unless you need to, use what the game provides if you can. But if you need to change an object, please do create a new one. Just rename an object and when a prompt comes up asking if you wish to create a new object, click 'Yes'. If you click 'No', the old object effectively no longer exists, and all references of that object take on the new name you gave it, plus any changes you made. Most likely this will cause problems. Not only that, but if you rename the object back to its old name, it just dirties your mod (you could do this, but you will have to clean this dirty reference out of your mod. You will clean your mod before you release it, right?). Keep it simple, and think before you rename objects haphazardly.
- **Naming New Objects.** When you name a new object use a unique name that no one else will likely use. Most modders put their initials at the beginning of an ID. If you use a name for an object in your mod that someone else also uses in their mod (same object or not), and someone loads both to play, they will conflict, probably screw up their game, and maybe cause a CTD. Here's an example: I create a new object from 'furn_crate_open_01' by renaming it 'mde_coins_odai_crate_01'. I used my initials (I can't use 'me' obviously, so I used 'mde'), then '_coins' for what it holds, '_odai' is where I put it, then '_crate_01' because it's the first crate I created. The key is to be original and name things in a way no one else will. If you need more examples, open up a mod and see what other people name their created objects. That should give you some ideas.

The Underscore '_': Placing this at the beginning of an ID causes the object created to be listed first on the tabbed list it falls on. Many people like to use it when renaming. But some people report problems with this (usually with scripts), other swear by it. It's up to you, but I myself don't.

***Note:** Do not open up the Stat Reference window of an object in the Render Window, and rename it. While it will only rename that singular selected reference (and create a new object), it will also put an asterisk on the original parent object, thus making your mod dirty (thus more to clean). Create a new object from the Object Window, delete the reference, then drop your new object in the space.*

- **Deleting Objects.** Don't delete objects unless it's user created. It's ok to delete a reference, assuming it isn't important to a quest or the game in general (and you better be sure about it), but there is no reason to delete an object based in the .esm file. If you want to delete an object *you* created, go right ahead. You can delete objects from an .esp, at worst it will screw up a plug-in.
- **Stat Window for Objects.** To bring up the statistics window for any object you can d-click that object, or r-click on it. R-clicking will bring up a menu with several options on it; New, Edit, Delete, etc. You do not get this stat window if you d-click on a reference, which will bring up the Reference Stat Window.
- **Reference Stat Window.** This window will be divided into two sections, the top and bottom, split by the Reference Bar. Do NOT make any changes to anything in the top section! This will make changes to ALL references of the parent object, to include the parent object! You *can* safely make changes to anything in the bottom section. Such changes will *only* affect the selected object.

- **Object Meshes and Textures.** Where are they? On the CS CD, unless you copied the CD to your hard drive. You can create a new object and search the CD (or your hard drive) for them. You can also make things easier for yourself and just rename an object (pay attention to above).
- **Adding an object.** To add an object to the world simply click and drag it to the Render Window where a reference will be created of it. If you need to add several objects, highlight those you wish to add and then click and drag the group of objects. Easy (of course now you have to sort and place said objects in the Render Window).
- **Blocked.** If this box is checked in an object's stat window the PC cannot interact or affect this item normally. If the PC tries to 'activate' an object that is blocked nothing happens. When the PC centers their view upon a blocked object, that object's name will still be displayed (this is commonly used for signposts, and a few other things).
- **Persistent.** If this box is checked on an object's stat window the object cannot be removed from the game by the PC, nor will it ever just disappear from the game world. Thus a dead actor will always remain where they died, an object sold to a merchant will remain in their inventory.
- **Name ID.** All objects *must* have a unique ID.
- **Name.** Almost all objects have a name. This is how the object is identified ingame by the PC. While all object IDs must be unique, objects can share a Name.
- **Weight.** Most objects, but not all, have a weight associated with it.

Let's get on to the objects. I will list them by tab.

Activator These are objects that do something, or allow the PC/actors to interact with. Many static like objects are created as activators so a script can be attached to them. Signposts, barriers, beds, etc. are all considered activators. You can attach scripts to activators. If you need a static object with a script to be attached, create it as an activator.

Mesh. The top button is for the mesh of the activator. Most are found under `Data Files\x` subfolder on the CS CD.

Alchemy These objects are potions. You can alter several values of a potion. You can also attach a script to a potion.

Mesh. The top button is for the mesh of the potion. These are found under `Data Files\n` subfolder on the CS CD.

Icon. The bottom button is for the icon in the inventory menu. These are found under `Data Files\n` subfolder on the CS CD.

Spell Effects. There are eight drop down menu boxes listing spell effects that can be added to a potion. These work the same as [Spellmaking](#). Note that the cost in the last column is the cost to purchase the potion, not the cost in magicka (using a potion does not cost magicka).

Apparatus These are the objects that are used to create potions ingame. You can attach scripts.

Mesh. Top button assigns the mesh of the object, found under `Data Files\m` subfolder on the CS CD.

Icon. Bottom button assigns the icon for the inventory. Found under `Data Files\m` subfolder CS CD.

***Note:** There appears to have been a couple of mistakes made. The Journeyman Calcinator is using the 'Tx_calcinator_01.tga' (used by the Apprentice Calcinator), but there is a 'Tx_calcinator_02.tga' that can be found on the CS CD. Did the wrong icon file get assigned? It's also using the same mesh, but there is a separate mesh for the Journeyman Calcinator. (did the developers forget their coffee that day?). Also, the Apprentice Mortar & Pestle is using the 'Tx_mortarpestle_j_01.tga' file (the same as the Journeyman Mortar & Pestle), but there is a 'Tx_mortarpestle_a_01.tga' file on the CS CD. Simple mistakes apparently, but easy to fix.*

Armor You must understand how armor is rendered in the game to grasp how to modify or create new armor. When the PC or an NPC wears armor (or clothing), the armor is not placed over the body part it is assigned to, it replaces that body part. The game is not rendering a body part 'under' the armor, only the armor. What this means is you must create or assign parts of armor under the Body Part tab (even if only one part), and then make (or assemble) the armor under this tab and assign the Body Part 'objects' created to the armor. This may be confusing, but if you look at a few pieces of armor already listed, it will make sense. Double clicking on an armor object opens a window to define stat values for the armor for use in the game. Before you can assign values to new armor, you *must* have already created it under the Body Part tab, or use those already created. The easiest way to create new armor types is to simply open up a piece of armor, change the object name of it (as explained in a bullet at the beginning of this chapter), and click 'Yes' when asked if you want a new object. If you want a full suit, make sure you do this for each piece of armor (i.e. a new type of steel armor? Make sure to create a new set of pauldrons, gauntlets, boots, greaves, a cuirass, helm, and shield). The sections and use of the window are:

Type. Drop down menu for what piece of armor it is (i.e. left pauldron, cuirass, right bracer, etc.).

Script. You can attach a script to the armor here.

Weight. The weight of armor. The game automatically assigns the weight 'class' of the armor based upon the weight you enter here. If you make the armor relatively light in weight, regardless of the AR you assign it, the game will treat the armor as 'Light Armor', whether you will it or no. If you make a piece of armor very heavy, the game will treat the armor as 'Heavy Armor'. Here is a table showing the weight classifications. Armor types are listed on the left, the maximum weight that can be assigned to that armor type and still be considered of a weight class is listed across the top.

Table 4.0 Armor Type Weight Allocation

	Light Maximum Weight	Medium Maximum Weight	Heavy Minimum Weight
Boots	12	18	Above 18
Cuirass	18	27	Above 27
Gauntlets/Bracers	3	4.5	Above 4.5
Greaves	9	13.5	Above 13.5
Helm	3	4.5	Above 4.5
Pauldrons	6	9	Above 9
Shield	9	13.5	Above 13.5

Health. This is the 'hit points' of the armor. Armorer skill and a repair object are used to restore these points. If a piece of armor is reduced to zero, the armor will become unequipped and useless.

AR. This is the value of the protection the armor gives. This number is set upon a skill level base of 30. This means that if the appropriate armor skill is 30, and the AR of a piece of armor is set to base 50, the armor has an ingame current value of 50. If the skill is lower than 30, then the value of the AR will also proportionally be lower. If the skill is above 30, then the AR value will also be proportionally higher. Try to assign values that are balanced.

Value. This is the monetary value of the item if bought or sold. Keep in mind you will pay more than this if buying, and will get less than this if selling.

Mesh. Upper button is clicked to assign a mesh to a piece of armor, found in `Data Files\A` subfolder on the CS CD. The mesh you need is the GND mesh, or 'ground' mesh. This mesh is what appears when the armor is dropped on the ground.

Icon. This lower button is clicked to assign an icon in the inventory menu, found in `Data Files\A` subfolder on the CS CD.

Picture box. This box shows the picture of the icon chosen.

Enchantment. This number is the maximum amount of possible enchantment points assigned to an armor piece (i.e. amount of magicka available for casting magic from the piece of armor), not the actual total. The actual points assigned is always 10 x 'enchantment cost'.

Enchanting. This drop down menu lists enchantments from the 'Enchantment' tab. Choose one to add to armor to make them magical.

Biped Object. These drop down menus are used to assign the parts that this armor covers. Note that some armor covers more than one body part (ie. boots cover both the left and right foot, and also the left and right ankle). Look at other armor types if unsure what to include. The body parts covered by armor type are:

Helmet	Head and Hair (some only the hair, not the whole head)
Cuirass	Chest
Pauldron	Forearm and Upper Arm (some also the clavicle/pauldron)
Gauntlets	Hands and Wrists
Greaves	Groin and Upper Legs
Boots	Ankles and Feet
Bracer	Wrists

Shield (Not assigned a body part)

Note: The above list **does not** apply to the Better Bodies mod. It has a different assignment system based upon it's single body mesh.

Male/Female Armor. If you intend this to be for a male to wear you must assign the appropriate part from the Body Part tab. Each drop down menu under this section must match the body part you listed to the left, under the male or female menu listing. Example: if you work with glass boots, you must list right ankle to one drop menu, then left ankle to the next below it, and then right foot, and finally left foot. Under the Male or Female section, to the right of 'right ankle' you must list the object ID of glass boots from the list either in the middle column for a male, or the right hand column for a female, for feet and for ankles (using the same object for both feet and for both ankles). If you list the object ID under both sections then either gender can wear this armor. If still confused, just look at another piece of armor of the type you wish to make or modify.

Body Part These objects are what make up Armor, a Body Part, or Clothing. However each is not composed of the same pieces. This will make sense as I explain it below. You cannot attach a script to an object in this window.

Part. This is the body part this object will replace when 'equipped' ingame.

Part Type. Click the object type for this object if creating a new one.

Female. If this is checked, the object will only be for females, if unchecked then it is for males.

Playable. If checked this object is available for the PC to 'wear'.

Mesh. Button assigns a mesh file to this object. Found under the Data Files\b subfolder on the CS CD.

Skin Info. This section is grayed out if working with clothing or armor. If you are working with a race, then the first drop down menu lists the race that this object is assigned (and only that race). The second drop down menu lists if the object is for a normal or vampire PC/NPC. This only comes into play when assigning a 'head' to a race. I will further explain some of the differences between armor, clothing, and race Part Types.

Armor. This is where you create the individual parts that make up a piece of armor. Or in simple terms, here is where you create the visible parts of armor that replace the PC's normal body parts. You must create each part here to create a full piece of armor as detailed above. This means that a pauldron must have a clavicle, upper arm, and forearm. These pieces are then assigned together as a single piece of armor and given stat values under the Armor tab. The Playable checkbox and the Skin Info section are both grayed out.

Clothing. This works basically the same as creating armor, except that different body parts go together under the Clothing tab to create most clothing (i.e. pants cover the ankles, upper legs, knees, and groin, while greaves only cover the upper legs and groin). The Playable checkbox and the Skin Info section are grayed out.

Race. This is where you assign meshes to body parts, creating a 'body' for the races defined by the Race Window. When creating or modifying a race you do not need to create or assign a 'clavicle' under part type (the clavicle is part of the 'chest' mesh). Only one body

part of each type can be assigned to a race and gender (i.e. only one male forearm, only one female upper leg, etc), except for head and hair. You can assign more than one of either (obviously). You must make sure a race is assigned if creating new race body parts from the drop down menu under Skin Info. This race *must* have already been created in the Race Window by clicking Character->Race. The easiest way to create a new race's body parts is to just rename all those given to a standard morrowind race (creating a new object), and then reassign the race for the new body part. Make sure to create both male and female body parts. If you try to mix different races you will most likely get clipping problems between the body parts. This is usually seen when someone creates a new race and mixes hair from one race to the head of another (ie. human hair on top of elven heads). Not to mention the textures won't match. The meshes for each are too different to match right so you see this as clipping in the game. Also, don't forget to assign a vampire head to a new race, or else when they become a vampire and the switch is made (or if you create a vampire NPC), no head will be visible, just the body with a big yellow exclamation point. This happens ingame whenever the game can't find a correct mesh for an object. There are quite a few plug-ins available out there on the net that provide new head and hair meshes for use in a game (see [Appendix B](#) for websites).

Book Books are easiest to modify or create by just renaming one and inserting your own text.

They can do a couple of other specific things. A 'book' can actually be a scroll or a book. Scrolls can also cast a spell (one time). You can attach a script to a book (or scroll).

Teaches. You can click this drop down menu and choose a skill from this list. The first time a copy of this book is read by a PC they will receive a one point increase in that skill.

Mesh. The upper button is used to assign this file. Found in the `Data Files\m` subfolder on the CS CD.

Icon. The lower button is used to assign the icon for the inventory. Found in the `Data Files\m` subfolder on the CS CD.

Scroll. Click this box if the object is a scroll. Make sure to use the correct mesh.

Enchantment. Enter a value if you wish a spell to be cast from the scroll. It must be enough to cast the spell you enter.

Enchanting. Click the drop down menu to attach a spell to a scroll for casting.

Text. You can enter text here for a book. There is a character limit of 64,000. Text used by the game in books is simplified HTML code, and must use only certain specific fonts.

Clothing The Clothing Window works exactly the same as the Armor Window (detailed above). The difference is what body parts are covered by types of clothing.

Shirt	Chest, Upper Arm, Forearm, Wrist
Skirt	Groin and Upper Legs
Pants	Groin, Upper Leg, Knee, and Ankle
Shoes	Foot and Ankle
Robe	Chest, Upper Arm, Elbow, Forearm, Groin, Upper Leg, Knee
Belt	(Not assigned a body part)
Ring	(Not assigned a body part)
Amulet	(Not assigned a body part)

***Note:** The above list **does not** apply to the Better Bodies mod. It has a different assignment system based upon it's single body meshes.*

Container These objects are barrels, chests, boxes, etc, and also many plants (yes, plants).

Containers cannot be picked up or moved. You can remove and place items in containers, unless it is a plant, which you can only remove objects from. You can add a script to a container. Some specific info:

Organic Container. If checked, this indicates the object is a plant, the PC will only be able to remove objects inside.

Respawns. This will be grayed out unless the organic container box is checked. If checked, objects inside the container will respawn about every 4 months or so.

Weight. The total weight of all objects within a container cannot exceed the weight value listed here. To the right is listed the 'encumbrance' so you don't exceed this value.

Animation. If this object has animation associated with it you can view the data by clicking this button.

Mesh. Click this button to assign the mesh and texture for the object. Found in the `Data Files\o` subfolder on the CS CD. There is no icon for such objects (though some modders have created containers that can be placed in the inventory and have icons for doing so).

Inventory List box. Here is where you put items to be found in the container. Just open another tab on the Object Window and drag and drop an object into the list. L-click once to highlight, and then l-click again (not a d-click) over the Count column to adjust the number of objects found in the container. If you enter a negative number the container will respawn the stated number amount of that item (which means they will always have that many, no matter how many times you remove the objects). Infinite items respawn after 24 hours.

Creatures The Creature Window is used to modify and create the beasts that annoy and chew on our PC during the game (ain't they cute when they do that). You can attach a script to a creature. If you plan on using a creature for a leveled list, but want just that creature on the list, create multiple versions of it at different levels. Make sure you give each a unique Name ID (the Name can be the same). Try to keep the balance of the game and your creature in mind. This Creatures Window has a lot of data in it, I will explain each carefully.

Type. Decide if the creature is a *Creature*, *Daedra*, *Humanoid*, or *Undead*. Daedra creatures cannot be harmed by normal weapons. Undead creatures cannot be harmed by normal weapons, and are affected by the Turn Undead spell effect.

Level. This is the level of the creature. This is 1 by default, the CS does not auto-calculate stats like it can NPCs for you. If you forget to change this, the creature will stay level 1.

Essential. If checked, and the creature is killed, the PC will receive a message the creature they just killed was essential to a quest, but not other info is given.

Corpses Persist. If checked, and the creature dies, the corpse will remain in the game world. The PC cannot remove it, nor will the game, after the specified period of time to do so for most corpses.

Respawn. If checked, the creature will be respawned the next time the cell is loaded.

***Note:** As best I can determine, if the PC does not remove the corpse of a creature killed, the creature will not respawn until the game removes the corpse.*

Attributes. These are the stats for this creature. You can set these as low as 1, or as high as 255. Most creatures should have a stat value between 1-100. Following stats are a little different.

Health. Defaults to $(STR + END/2) + (Lvl * END/10)$. You can change this manually. If you set the Health to zero, the creature will be set as a corpse when the cell loads (they still start in an upright position, but fall over upon cell load). Maximum possible Health is 65535.

Spell Points. Defaults to $(INT * \text{ability multiplier})$. You can change this manually. Maximum Spell Points is 65535.

Fatigue. Defaults to $(STR + END + AGL + WIL)$. You can change this manually. Maximum Fatigue is 65535.

Soul. The value of their soul for charging a soul gem. If you set this to 400 or more, the creature's soul can be used to create Constant Effect enchantments. Maximum Soul is 99999.

Skills. The creature only has three skills, *Combat*, *Magic*, and *Stealth*. This is the skill level used for any skill from one of those categories. You can set these manually. Maximum skill level is 255.

Attacks. Creatures can have three different attacks, you must set the minimum and maximum amount of damage for each.

Blood Texture. *Red* for most creatures, *White* for skeletons and some undead, and *Gold* for metal sparks.

Sound Gen Creature. Drop down menu listing of creatures. Pick the creature to derive sound from.

Dialog. Creatures do not have any dialog by default. If you choose to add it click this button to open the [Dialog Window](#), filtered to this creature.

Animation. Opens an animation data window for this creature.

AI. Opens the AI window for this creature. There are two tabs; **Packages** and **Services**.

Packages. This tab allows you to set the AI Packages that dictate how the creature acts and reacts in the game world. See NPCs for detailed information on [AI Packages](#).

Services. Creatures normally do not offer services. However, you *can* set them to do so. See NPCs for detailed information on [AI Services](#).

***Note:** If you have a creature set to offer services, any buying/selling is done at base value of the items/service. Creatures do not have a Mercantile skill, nor are affected by it.*

Mesh. Click to assign a mesh/texture to a creature. Unlike NPCs, you do not assign every body part, only one mesh file is used for a creature. These files can be found in the `Data Files\r` subfolder on the CS CD.

Scale. This defaults to 1.00 (normal). The creature can be set to half of its normal size (0.5), or twice it's normal size (2.0). [Scripting](#) can be used to set Scale beyond these limits however

***Note:** Scale can also be set for a creature (or any object) within the Render Window by holding down the 's' key, then l-click and drag the 'creature' up or down.*

Weapon & Shield. If you check this box, and you check the box for Biped animations for movement, the creature will use the best weapon and shield you include in their inventory (if you give them any).

Movement. Click on the type of primary movement in this section. Make sure you have a sound assigned to appropriate actions for movement (unless you want them to be silent as they move). If you click on *Biped* the creature will use the animations for NPC movement, and fly, walk, and swim will gray out. If you set movement to *None*, other movement will gray out, and they will not move.

Encumbrance. Creatures are affected by encumbrance too. Shown is minimum/maximum for a creature according to attributes.

Item Inventory tab. This box shows items the creature carries. To add an item or a leveled item list, just click and drag it into this box. To remove one, highlight it and click 'delete'.

Spell Inventory tab. This lists all spells the creature can cast. Add or delete spells as you would items.

Door Doors are just that, doors. You can attach a script to them. When you place them remember you need a doorframe for them to fit into.

Animation. If animation is associated with the door object you can view it here. Grayed out if it has none.

Mesh. This button is to include the mesh. Found in the `Data Files\m` subfolder on the CS CD. There is no icon button.

Sound Open. Click on the middle button to assign a sound for when the door is opening.

Sound Close. Click on the bottom button to assign a sound for when the door is closing.

Enchanting The Enchanting Window works exactly the same as the Spellmaking Window. An enchantment is added to another object so that this other object is magical, and can 'cast' a spell. Thus a PC needs no knowledge of the spell nor any magicka to cast it. The Enchanting Window has a few differences from the Spellmaking Window.

Cast Type. This describes how the enchantment is cast, or when.

Cast When Used. This means the enchantment will only be cast when the object it is placed on is activated. Usually this involves equipping the object if in the PC's inventory.

Cast When Strikes. This can only be placed on weapons. When a successful strike is made this enchantment is cast.

Cast Once. This can only be placed on scrolls. The enchantment is cast once when the scroll is used.

Constant Effect. The enchantment is 'always on', it does not need to be cast. The object only needs to be equipped and the enchantment takes affect.

Charge Amount. This is the minimum magicka pool that an object will have to cast the enchantment. For Cast Once or Cast When Used, this is equal to the Cost. For Cast When Strikes this is 10xCost (or 10 successful strikes). It is grayed out for Constant Effect.

Auto-Calculate. Check this box if you wish the CS to total the cost of all spell effects for you. Uncheck and you can enter any amount you wish in the Cost box.

Cost. This is the cost of casting this enchantment. This is the amount subtracted from the magicka pool assigned to an object for spell casting. This is grayed out for Cast Once and Constant Effect.

Ingredient These objects are able to instill a limited spell effect if eaten or used, or can be mixed with other ingredients to create potions. You cannot determine the amount of an effect, this is determined by the PC's Alchemy skill. You can attach a script to an ingredient.

Mesh. Click this button to assign the mesh/texture. Found in the `Data Files\` subfolder on the CS CD.

Icon. Click this button to assign an icon for the inventory. Found in the `Data Files\` subfolder on the CS CD.

Effects. Up to four spell effects may be assigned to an ingredient, one for each drop down menu. Only the first one listed will have any effect if eaten.

Leveled Creatures This list acts as a reference in the game world. When the cell loads the PC's level is checked against this list. If a creature is on the list that meets the parameters as set on this list for the PC, that creature is added to the world at the reference's grid. If no creatures on the list meet the parameter, no creature appears.

Calculate From All Levels<=PC's Level. If this box is not checked, only creatures that are equal to the level of the PC will be placed at that location. If checked, any creature equal to or lower in level to the PC will be placed at that location. (If more than one creature meets these parameters, one is chosen at random).

Chance None. This is the percent chance that no creature will be chosen when the cell loads.

List box. This is the listing of creatures. Drag and drop creatures (or an NPC) into this box. They will load top to bottom, and cannot be shifted about in the list. Each creature listed must be set to a level equal to or higher than the creature above it. This means add creatures highest level first, lowest level last. You can add another leveled creature list in this box (no kidding, you can, it is considered a creature. Just give it a level, and if chosen, that list picks randomly from any creature within it, regardless of level associations).

PC Level. This is the PC's level at which you want this creature to appear. This must be equal to or higher than the creature above it. This defaults to 1, but can be adjusted manually. To do so l-click once and highlight a creature, then l-click again on the list and enter a level you wish.

Creature Name. This is the Name ID of a creature.

Leveled Items This works similar to the leveled creature list. The leveled item list is placed within a container or on a creature/NPC. When the host object is activated, or opened, the game checks the PC's level and places an item, or items, accordingly. If no item meets the parameters, no item is placed.

Calculate From All Levels<=PC's Level. If this box is not checked, only items that are equal to the level of the PC will be placed at that location. If checked, any item equal to or lower in level to the PC will be placed at that location. (If more than one item meets these parameters, all items will be placed).

Calculate For Each Item In Count. When checked, the list calculates the Chance None percentage for each item listed.

Chance None. This is the percent chance that no item will be chosen when the object is activated.

List box. This is the listing of items. Drag and drop items into this box. They will load top to bottom, and cannot be shifted about in the list. Each item listed must be set to a level equal to or higher than the item above it. This means add items highest level first, lowest level last. You can add another leveled item list in this box (just like leveled creature lists above).

PC Level. This is the PC's level at which you want this item to appear. This must be equal to or higher than the item above it. This defaults to 1, but can be manually adjusted. To do so l-click once and highlight an item, then l-click again on the list and enter a level you wish.

Item Name. This is the Name ID of an item.

Light An object that gives off light. The light does not have to be a physically seen item, it can just be a source of light. It can also be an item that can be picked up and carried. You can attach scripts to lights.

Radius. The radius in units the light shines on objects. Thus highlights and shadows will only be created for objects that fall within this radius.

Can Carry. Check this box if the object can be picked up and carried. Make sure an icon is assigned.

Off By Default. This box is normally grayed out. If you check the Can Carry box, this box becomes available. If this box is checked for an object that can be carried, when the PC puts the object down (drop) the item will not 'light up'. If this box is not checked, and an inventory Light object is dropped, that object will begin emitting light (if you put a candle down it will start to burn). If checked, the item must be equipped to begin emitting light (like a torch).

Name. The CS help file suggests you only name lights that can be picked up. A good idea, otherwise every time the PC centers on a mushroom giving off a little bit of light, he will be given the name of the mushroom AND of the light object.

Icon. If the item is checked for Can Carry, you click this button to add the icon for the inventory. Found in the `Data Files\1` subfolder on the CS CD.

Picture box. Shows a picture of the icon you have chosen. Grayed out if object can't be carried.

Time. Number of seconds the light will burn when equipped by the PC, and only the PC. Grayed out if the object can't be carried.

Fire. If this box is checked, the light is designated a fire by the game, and particle effects are added.

Negative. If checked, the light does not emit light, but removes it. Example: you create a white source of light and check this box. When the light is lit, instead of a white color of

light emitted in the defined radius, all white light is removed from within the defined radius.

Dynamic. If checked, the game will apply lighting affects to moving objects. The difference is; for the small glowing mushroom little light is emitted and does not highlight nor cause shadows on moving objects (like the PC, NPCs, or creatures). For a burning torch that flickers (dynamic), or a lantern, the light given off highlights features and causes shadows to be given off by things that move (PC, NPC, or creatures).

Sound. If the light should give off a sound, add the sound file by clicking this button and browse to the file to associate it.

Mesh. Click this to assign a mesh. If the light object is not an item, don't assign anything. Found in the `Data Files\n` subfolder on the CS CD.

Color. A standard color palette. Choose the color you want the light to be.

Flicker Effects. These are checked if you wish the light object to vary it's brightness.

Flicker. For a varied flickering effect such as that given off by a torch or campfire.

Flicker Slow. The flickering effect is slowed down.

Pulse. For a constant altering change in the light intensity.

Pulse Slow. For a slower constant altering change in the light intensity.

Lockpick These objects are used to pick the lock of doors and containers (if locked). You can attach a script to these objects.

Quality. This value is a multiplier that works like this; assuming a skill of 50 with a lockpick quality of 0.5, would give you an effective skill of 25 ($50 \times 0.5 = 25$).

Mesh. The top button will assign a mesh/texture. Found in the `Data Files\m` subfolder on the CS CD.

Icon. The bottom button will assign an icon for the inventory. Found in the `Data Files\m` subfolder on the CS CD.

Misc. Item These are objects that don't necessarily fit under any other tab. Keys fall under this tab, as do visual effects for spell effects (see [Spell Effects](#)). You can add a script to these objects.

Mesh. Top button assigns mesh/texture for object. Found in the `Data Files\m` subfolder on the CS CD.

Icon. Bottom button assigns an icon for the inventory. Found in the `Data Files\m` subfolder on the CS CD.

NPC NPCs can be just as robust and detailed a character as the PC. Scripts can be added to an NPC. A great deal of information needs to be assigned for an NPC. For an average Joe it may not require much work, but for an NPC that is a focal point of a quest you may need to do quite a bit of thinking before you actually sit down and modify or create them in this window.

Race. Click the drop down box and choose which race your NPC will be. Unlike the PC, an NPC can be *any* race that has been created. The appropriate body part files will be added in the top right two boxes, you do not need to add anything there. The boxes will list the available head (upper) and hair (lower) files you can choose from to give your NPC.

Female. Click this box if the NPC is to be female.

Class. Click this drop down menu and choose from any class. If you want a customized class you must first create it.

Level. Enter the level of your NPC. This defaults to 1.

Faction. Click this drop down menu to choose the faction your NPC may belong to. If none, just leave it blank. Once chosen, click the drop down menu to the right and pick their rank.

Essential. If checked, and the NPC is killed, the PC will receive a message the NPC they just killed was essential in the game.

Corpses Persist. If checked, and the NPC dies, the corpse will remain in the game world. The PC cannot remove it, nor will the game after the specified period of time to do so for most corpses.

Respawn. If checked, the NPC will be respawned the next time the cell loads.

***Note:** As best I can determine, if the PC does not remove the corpse of an NPC killed, the NPC will not respawn until the game removes the corpse.*

Attributes. These are the stats for this NPC. NPCs should have a stat value between 1-100. Maximum is 255.

Health. Defaults to $(STR + END/2) + (Lvl * END/10)$. You can change this manually. If you set the Health to zero, the NPC will fall over dead when the cell loads. Maximum Health is 65535.

Spell Points. Defaults to $(INT * ability\ multiplier)$. You can change this manually. When these are gone the NPC will only use physical attacks. Maximum Spell Points is 65535.

Fatigue. Defaults to $(STR + END + AGL + WIL)$. You can change this manually. Maximum Fatigue is 65535.

Disposition. Defaults to 50. You can change this manually. This will be constantly modified ingame based upon the PC's actions, race, class, faction, etc. Maximum Disposition is 100.

Rep. Defaults to 0. You can modify this manually. The higher the reputation of an NPC, the harder it is for the PC to persuade them. Maximum Reputation is 255.

Blood Texture. *Red* for most creatures, *White* for skeletons, and *Gold* for metal sparks.

Auto-Calculate Stats. Check this and the CS will calculate all stats, skills, and spells based upon race, class, and level. They will be grayed out. Uncheck to modify them manually again.

Skills. In this box are two columns. To the left is the *skill level*, to the right is the *skill*. This is set by default according to class and race. You can adjust the skill level; l-click once to highlight, and then l-click again on the skill level. Enter the value you want for a skill. Maximum is 255.

***Note.** For Armor and Weapon skills: an NPC will use the weapon and armor in their inventory that they have the highest skill in. This means if you give them an Unarmored skill level of 50, and Light Armor of 30, then give them light armor to wear, they won't wear it. This is because their Unarmored skill is higher, so they will use that skill. The*

same goes for weapons. They will use whatever weapon in their inventory that they have the highest skill rating with, regardless of what you want them to use.

Dialog. Clicking on this will bring up the Dialog Window filtered for this NPC (see [Dialog Window](#) for more info).

Animation. Opens up the animation file used by this NPC so you can see the data.

AI. Opens the AI window for this NPC. There are two tabs; **Packages** and **Services**.

Packages. This tab allows you to set the AI Packages that dictate how the NPC acts and reacts in the game world. A Wander package is present by default. Click the drop down menu to add a new AI Package. Highlight a package in the box and click 'delete' to remove it.

Fight. This is how much the NPC wants to fight the PC. Default is 30. Many actions of the PC will alter this ingame. Below is a table pulled from the CS Help file showing an NPC's actions based upon their *Fight* value.

Table 4.1 Fight Value Reactions

100	Always Attacks
95	Will Attack as PC gets close (3000 units)
90	Will Attack as PC gets close (2000 units)
80	Will Attack as PC gets close or if he dislikes you (1000 units, 40 Disp)
70	Will Attack if close and strong dislike (1000 units, 35 Disp)
60	Will Attack if he dislikes you and you get close (Disp below 30)
50	Will Attack if he hates you (Disp at 0)
40	Will attack if he dislikes you, and you get close. (500 Units, Disp 10)
30	Will Attack if hates you and you commit crime.
20	Will Attack if dislikes you and multiple crimes.
10	Will attack if he hates you and you do multiple crimes on him.
0	Will ONLY attack if attacked first.

Flee. This is how likely the NPC is to flee from a fight. Default is zero. Many factors go into this formula of which I have no details. Basically put, if the NPC is not able to put out much damage, and the PC is much more powerful, the more likely the NPC will flee a battle than fight, or continue to fight. A value of zero will cause the NPC to fight to the death.

Alarm. This is the chance an NPC will react to seeing a crime committed. If seen, an NPC will *shout* at the PC and other settings will go up or down (Disposition, Fight, etc) based on the crime. This will also alert other NPCs, whose settings will also go up. If an NPC hears the shout and has an Alarm setting of 100, they will put money on the PC's head. This value is based on the crime committed as shown below as taken from the CS Help file.

Table 4.2 Monetary Results Of Crimes

PC Crime	Gold
Attack NPC	40

Killing	1000
Stealing	1
Pick Pocketing	25
Trespassing	5
Taunting	5
Intimidation	5

Note: If an NPC's Class is 'Guard', they will also run after the PC and attempt to intercept them, and then initiate dialog (make them pay the money on the PC's head as a 'fine', or arrest them and they go to jail), **or** if the PC's crime level is over 10,000 they will initiate combat.

Hello. This default is set to 30. This is the distance at which an NPC will stop and greet the PC. This number is multiplied by the iGreetingDistanceMultiplier game setting (which is by default set to 7), thus 210 units (or just under 10 feet) at a rating of 30.

Services. This is the ability of an NPC to buy and sell goods within their inventory, offer training, spellmaking, enchanting, or repair work. Simply check the appropriate boxes. You can check as many boxes as you wish, even all of them (Die Uber NPC).

Buy/Sell. Check the box of the type items you wish the NPC to merchant in. They will only buy/sell items of that type listed within their inventory. Once an item is sold, it is gone and is not restocked. However, if you set the number of items in their inventory to a negative number, that number in items will respawn instantly, for that item, upon closure of the barter menu by the PC. Enter the amount of Gold the merchant has available in the Barter Gold box. This amount will reset after a specified amount of time (but not sure how long). If you set this to a negative number, the merchant will have this amount of gold 'respawn' every twenty-four hours.

Other Services. These are Training, Spellmaking, Enchanting, and Repair. You cannot alter the amount of money charged for these services, it is set by the game.

Training. The NPC will train the PC in the three highest level skills they have. They cannot train them above their own skill level.

Spellmaking. Allows the NPC to make new spells for, and sell spells they know to the PC.

Enchanting. Allows the NPC to create enchanted items for the PC.

Repair. Allows the NPC to completely repair armor and weapons for the PC.

Travel Services. This allows the NPC to grant fast travel options to the PC. This is accomplished through dialog. However, you must set the destination cell with the first drop down menu list. Then click on the Select Marker button to the right and find the marker to link to for this choice of travel. Only four destinations can be offered maximum.

Cell. Click this drop down menu to choose the destination cell you wish to offer as a choice to the PC.

Select Marker. Click this to go to the cell chosen. The cell will load and a marker will appear centered in the cell. Click and drag the marker within the cell to where you want the PC to appear. The direction of the arrow will be the

direction the PC faces when the cell loads ingame and they appear. It doesn't have to be perfectly at ground level, the game will place the PC on the ground.

Note: *For some reason every time I hit this button it put the marker within a rock or some such object. I had to spin around to find it, move it so I could see it properly, then place it. Within interiors it always seems to put markers at the same exact grid as the Northmarker (I have to move the Northmarker to find and move my Travel Marker. Very annoying).*

Return. Clicking this will return you to the cell your NPC is in (or the last loaded cell).

Keep in mind: *Once you create a Travel Marker through travel service, you cannot delete the Travel Marker in the Render Window, period. You must find the NPC that has service to that Travel Marker, and then set the destination in 'Cell' to None. The marker will then disappear. If you delete an NPC that has travel service, any markers they provided travel to will remain, and you will NOT be able to delete them. I know of no utilities that will enable you to delete them either, and believe me I've tried.*

Probe These objects are used to disarm a trap. They operate the same way as the lockpick objects work. You can attach a script to these objects.

Quality. This value should work exactly as it does for lockpicks.

Mesh. The top button will assign a mesh. Found in the `Data Files\m` subfolder on the CS CD.

Icon. The bottom button will assign an icon for the inventory. Found in the `Data Files\m` subfolder on the CS CD.

Repair Item These object are used to repair armor and weapon's health value when it falls below maximum. They operate the same way as the lockpick objects work. You can attach a script to these objects.

Quality. This value should work exactly as it does for lockpicks.

Mesh. The top button will assign a mesh. Found in the `Data Files\m` subfolder on the CS CD.

Icon. The bottom button will assign an icon for the inventory. Found in the `Data Files\m` subfolder on the CS CD.

Spellmaking These objects are spells, curses, diseases, and spell like abilities that exist in the game. Spells and such are composed of [spell effects](#). Spell effects cannot be directly used in the game to create magic, a spell is created and the spell effects are added to it to define what the spell does. You cannot add a script to a spell. To create a spell you must decide what spell effects to include in the spell (what it does), the parameters of those effects (how it does it), cost (in magicka), and the result.

Effects. There are two drop down menus. The spell effects that define what a spell does are picked from the first menu list. You can add a maximum of eight spell effects to a spell. They can be any effect, each with it's own defining parameters, within one spell. A few spell effects can be further defined (ex: Absorb skill; you must pick which skill is

absorbed from this menu list). These are added in the second menu list (which normally is grayed out).

Cost. These are the parameters for each spell effect chosen in the drop down menus to the left. The spell parameters act as multipliers in figuring out the cost to cast a spell created. This is not set in stone however, you can assign a cost different than what the CS shows. Not every spell will need every box filled, in which case it will be grayed out.

Range. Click to pick either Self, Target, or Area. Self is whoever casts the spell. Target is whomever the spell is cast at, other than the one who casts the spell, as long as they are in [Line Of Sight](#). Area means it affects the target plus a distance in radius outward.

Area. This will be grayed out unless you choose 'Area' for Range. The number entered represents radius distance in feet (i.e. a value of 5 = 5 foot radius).

Duration. This is how long in seconds a spell effect will last. A duration of zero means instantaneously. Some spells can only be instantaneous.

Magnitude. This is the degree to which the effect impacts the target. For some effects it impacts numerically in Health, an Attribute, Fatigue, etc. Others it represents distance, as in sEffectDetectEnchantment (number = feet). Others it is the level of a creature affected, as in sEffectCommandCreatures. These are described in the (very) small manual included originally with Morrowind under spells.

Type. This is what form the spell takes. This affects how it is applied to the object cast upon. The types are:

Spells. Spells have a cost to cast, and may fail depending upon the skill level of the PC. The PC may create new spells, and NPCs can sell them.

Ability. An Ability is usually given to a race or creature (usually) from the Race or Creature window. An Ability is a *constant affect* set to the parameters you give any spell effects listed. An Ability cannot be sold, is not cast (it's always on), and does not need to be learned. It usually reflects a natural ability or knowledge inherent to a race.

***Note:** Abilities can be added and removed with a script, except those added due to Race or Birthsign.*

Blight Disease. Blight Diseases are like a constant effect spell, cannot be dispelled, nor be cast. They can only be 'removed' by *curing* them with a cure blight disease spell effect.

Disease. Common diseases. Similar to Blight Diseases, but not as debilitating, and simpler to cure. A PC can usually cast cure common disease spell effects, as they are cheaper and easier to cast than a cure blight disease spell effect.

Curse. A curse is a special type of spell that works like the 'disease' spells. It takes a remove curse spell effect to remove a curse.

Power. A power is a spell that a PC, NPC, or creature may cast once per day. They usually don't cost magicka, but a cost can be assigned. These always succeed in being cast.

Auto-Calculate. Check this box and the CS will calculate the cost of a spell automatically.

PC Start Spell. Check this box if you want all PCs to start the game with this spell.

Always Succeeds. If checked, the spell will always succeed when cast. You only need to use this with spells; Abilities, Diseases, Curses, and Powers always succeed.

Spell Cost. This is where the cost (if any) is figured and must be displayed for all included spell effects. You can assign any cost here you wish to, even no cost (zero).

Static These objects are usually things like rock, trees, walls, etc. You *cannot* attach a script to these objects.

Mesh. The top button will assign a mesh/texture. Found in the `Data Files\i` subfolder on the CS CD. There is no icon as they cannot be put in the PC's inventory.

Weapons Objects that are weapons. You can add a script to a weapon (except throwing weapons).

Type. Click the drop down menu to choose the type of weapon.

Health. This is the 'hit points' of a weapon. Armorer skill and a Repair Object are used to restore these points. When the health reaches zero the weapon will become unequipped and useless until fixed.

Speed. How fast the weapon is swung. Default is 1.00, thus a value of 2.00 would be twice as fast, while 0.5 would be half a fast.

Enchantment. This is the maximum potential magicka pool a weapon can have.

Reach. This is how far away from the PC the weapon reaches to strike something. Defaults to 1.00 (normal for a longsword). This rating is not used for Ranged Weapons (arrows, darts, etc).

Enchanting. Click this drop down menu to see a list of all enchantments from the Enchantments tab, choose one to add to a weapon.

Ignores Normal Weapon Resistance. If checked, this weapon can hit any NPC or creature with the Resist Normal Weapons spell effect active, either as a spell or an ability.

Mesh. The top button assigns the mesh/texture for the weapon. This file can be found under the `Data File\w` subfolder on the CS CD.

Icon. The bottom button assigns the icon for the inventory. This file can be found under the `Data Files\w` subfolder on the CS CD.

Damage. Any weapon will have three methods of inflicting damage; Chop, Slash, and Thrust. Assign the damages you want the weapon to do depending upon the attack form made. If unsure how you want the weapon to do damage, just look at similar weapons.

***Note:** Ranged weapons only use the Chop value for damage.*

Silver. If checked, this weapon is made from silver and will damage those creatures/beings vulnerable to silver.

The Render Window

The Render Window shows the game world while modifying, adding, deleting, or just plain looking at it. Moving about the 'scene' in the window can take some getting used to, learn to use the keys on your keyboard with the mouse to move about and rotate what you are viewing. The main function of the Render Window is to view the world as seen in the game, add/move/delete

objects in the world (references), and/or to modify the landscape. There is not a whole lot to say about the Render Window, but here are some tips to keep in mind when working in it:

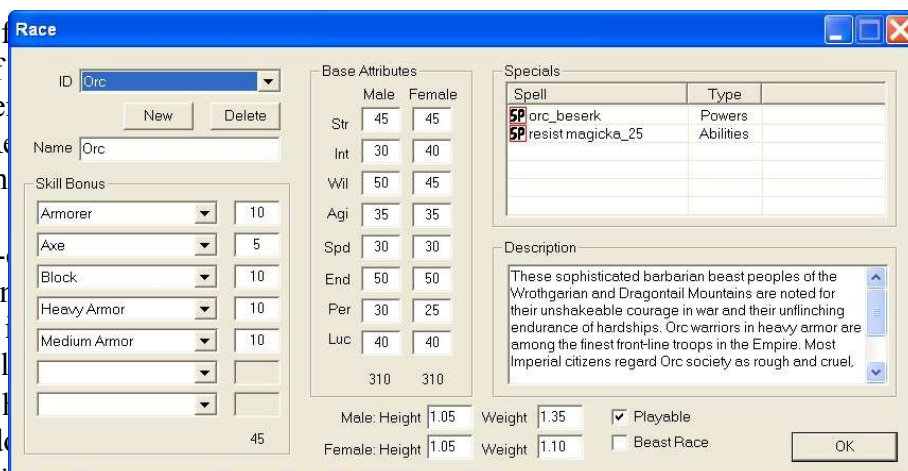
When viewing:

- The Render Window is for viewing cells and references placed in it, use the Preview Window to view objects from the Object Window.
- Movement may take some getting used to. Check out the [Shortkeys](#) to help you understand movement about the Render Window.
- Use [Brightness](#) if the screen is dark, but don't forget to add lighting if you do. I have a tendency to use brightness and forget that the cell I was working on was still dark ingame.
- When landscaping, try doing it in [Wireframe](#) mode. I found it easier on my eyes once I got used to it, and my system rendered changes quicker. I don't suggest vertex painting or texturing in wireframe mode (though you can).
- Don't forget that if your movement about the window seems slow, you can adjust the speed or amount of movement under [Preferences](#). Pressing the up, down, left, right arrow keys will move you north, south, west, east direction exactly one half a cell length. But watch out, it will move those directions regardless of which direction you are facing.
- You can place an object so that only part of it is visible (sinking a boulder into the ground so that only a part is visible creates a small rock). This is fine and useful. However, the game will continue to render the entire boulder, even the part that will never be seen. Keep this in mind as you build. Just because it doesn't seem like much is in a cell, unseen polygons could be causing a drop in FPS.

Objects:

- Selected objects appear to have a 'box' of red and green lines around it.
- Not sure which way is North? Select an object and hit 't'. You will move to a top down view over that object and the Render Window will orient so that north is now 'up' in the window. When in an interior cell always place a [Northmarker](#) (suggest you make it your first object placed). This will help keep you oriented and the game likes it.
- Northmarkers always point North. When you place one, place it so that it points the direction you *want* to be north, not what you *think* is north.
- Remember: when you place a reference (of an object) you can make it bigger/smaller, or rotate and move it about, but do not change the *stats* of the object. If you do, you are changing the stats of that reference's object. This changes EVERY reference of that object in the game. If you need to change the stats, make a new object first in the Object Window (by renaming it), and use that new object. Refer to [The Object Window](#).
- If you are having trouble placing an object exactly where you wanted it, try using Angle Snap and/or Grid Snap for big statics. For smaller items you will find it easier to keep it off for small movements and/or reference placement.
- If placing several references of the same object in a window, it may be easier to place a reference and then [duplicate](#) it. You can then place the duplicates (which stack upon the original reference) where you want. You can also use the keyboard shortcuts of ctrl + x/c/v to cut/copy/paste objects. You can ctrl + l-click to select/deselect objects (even as a group).

- When you place your reference in the Render Window (an oddity of the software), double click the reference to bring up the Reference Data Window centered on it in the Render Window. Change the 'x', 'y', and 'z' coordinates to find.
- To *move* a reference in the Render Window, click and drag movements. R-click and drag to move vertically, or if just moving horizontally, left or right, only horizontally.
- To resize a reference, click and drag the corners, or you can manually do it in the Reference Data Window.
- Even though many objects are intended for use in an exterior or interior cell, almost any object can be placed in either, regardless of the original intent. Nice huh.
- If you plan on adding a script to an object, *make a new object* and add the script to it. This will help keep your plug-in simple to work with, and compatible with other mods.



Building:

- When building interior cells and putting buildings together in exteriors, use [Angle Snap](#) and [Grid Snap](#) when placing a reference. Make adjustments for the amount of 'snap' that work for you. This will make building much easier and quicker than trying to line objects up seamlessly manually, especially when you first learn to do this.
- When building, remember the building blocks (walls, corners, etc) are references of objects too. Treat them as such.

Object Reference Window

When you double click on any object in the Render Window it will bring up the Object Reference Window. This window will provide all the data about that reference. Some objects will have more data than others. There are two parts to this window, that data which is above the 'Reference Data' line, and that which is below it. All the data which is listed above this line pertains to ALL references of an object. You change anything here and the change will be applied to all references of this object, including the parent object. This is a no-no! To change parent object data, find it in the Object Window and change it there! All data listed below the Reference Data line is specific to only this reference you clicked on. Depending on the object some, none, or all of the boxes may be grayed out. This is where you assign an object ownership (example: merchants can 'own' any object you assign to them that's not on their person but sitting in their shop. PC tries to pick up object or pick the lock of such, and they're busted if seen). Assign an object as locked, set the level, assign a key to unlock it, or add a trap to it. Assign a soul to a soul gem, or even add a global/variable to be associated with an object.

The Preview Window

The Preview Window by default is not active, the Render Window is. To open it simply click on the View Menu and then Preview Window. This window is used to view objects. It is not for landscape, references of objects, or to view anything in the game 'world'. If an object has animation associated with it the object viewed will be shown animated (usually), and you can

click on the buttons to cycle through the different animations. If it has sounds directly associated, you'll hear them. Viewing can be awkward, if you have this window small you may not be able to view or find an object selected. Simply expand the window, click a different object (so it refreshes), and use the arrow keys (or page up and page down) to find or view an object. The arrow keys will move an object by rotating it within the window to get it to a position so that you can view it. This can be tricky for some objects. The functions listed below the grayed out Current Cell Only under the [View Menu](#) have no function in the Preview Window.

The Cell View Window

The Cell View Window is comprised of two parts: The left side that lists cells and the right side that lists contents of those cells.

- Single click once in the Cell List (left side) on a cell to see all references (of objects) within it listed in the Reference List (right side). D-clicking a cell in the Cell List will load the cell in the Render Window along with all references in the Reference List.
- A single click will highlight a reference in the Reference List (right side). A d-click will load the cell that reference belongs to in the Render Window, and Center your viewpoint upon that selected reference.
- You can click upon the top of any column to change the listing under it to be listed alphabetically, or numerically, ascending or descending.
- D-clicking upon a reference in the Reference List will not bring up that reference's Stat Window, it will merely load the cell and center you upon that reference. *Then* you can d-click the reference in the Render Window and bring up the Reference Stat Window.

Cell List (left side). You can scroll through all cells in the game.

Name. This shows the names of a cell.

Grid. This shows the grid of the cell.

Ref Count. This number is the amount of references of objects that exist within the cell.

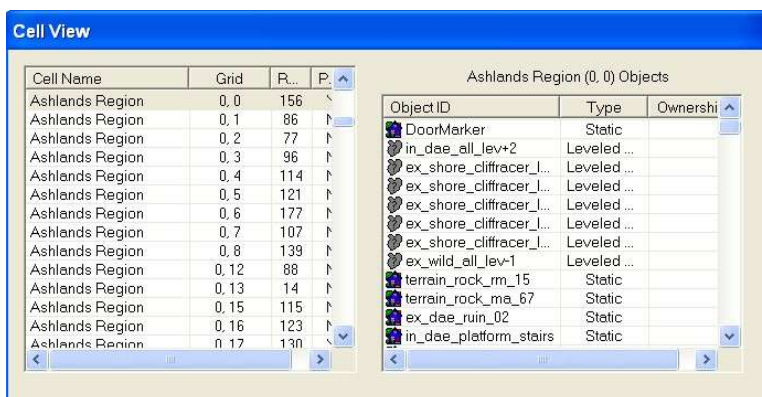
Path. Does the cell have a Path Grid within it (yes or no).

Reference List (right side). This shows the cell name and grid at the top.

Object ID. This is the object ID of the reference.

Type. The Object Window 'tab' the reference can be found under.

Ownership. What actor this reference belongs to.



The Race Window

This brings up a window within which you can edit, create, or delete a race. This does not create, edit, or delete a *body* for a race, this only defines the data about a race. Let's look at each box of data.

***Note:** Do not make changes to the provided races without some forethought. A race should be treated just like an object, you change one thing about a race in this window and you change every reference of that race in the game. This could have unforeseen consequences.*

ID. This drop down box displays the object name for a race. The game treats a race as an object, so the object naming conventions still apply. The ID does not show up in the game.

Name. This is the name by which the race will be known ingame.

New. Clicking on this will bring up a separate box to type in a new ID Name. When you hit 'ok', the box disappears and the Race Window is now blank with your new race ID name in the appropriate box.

Delete. To delete a race you don't want, click on ID Name, scroll to find the race's ID Name and click on it. That race's statistics should be displayed. Click on the "Delete" button. You will be prompted to verify you want to delete it. Do so. The race's ID Name will then show a "D" to the right of the ID name. The race is not truly deleted until the next time you hit the save button.

Skill Bonuses. These are the skills a race receives a bonus to in the game. There are seven drop down menus, with a small box to the right of each. You can assign up to seven skills for a race. Click on a box to see all the skills. Pick one and it appears in the box. Now choose a number for the bonus for that skill in the box immediately to the right. This can be any number up to ten digits, but will ignore the number if it goes above ten digits (not sure how the game would handle a big number). The CS help file states the total from all bonuses should not exceed 45 for balance, but this is up to you.

Base Attributes. These are the starting attributes of Strength, Intelligence, Willpower, Agility, Speed, Endurance, Personality, and Luck. You must assign a value to both males and females (male on the left, female on the right). You can assign a starting value of up to 255 for a race. The 'normal' starting value is 30-40. The CS help file suggests a starting total of all attributes to equal 310. Again, this is up to you.

Specials. Here you can add or delete spells, powers, and abilities to a race, or change the ones they have. To add, open the Object Window (do not close the Race Window). Click on the [Spellmaking](#) tab. Choose a spell from the list or make a new spell (see [Chapter 4](#)). To add a spell to a race click and drag the spell to the magical abilities box. It should then appear in the box. You can highlight more than one spell and drag all as a group. To edit a spell, you do not click on the magical abilities box in the Race Window, this only displays abilities for that race. Instead edit the spell itself under the Spellmaking tab. To delete a spell ability from a race, merely highlight that spell in the spell abilities box, and then hit your 'delete' key. This will remove the ability from that race, but will not delete the spell (that is done from the Spellmaking tab).

Description. In this box is provided a description of a race. You can have this say whatever you want. When the Stat menu is open ingame and you hold the mouse over the Race Name, this text will appear as a floating text box.

Height/Weight. There are four boxes shown here. Two to the right of 'Males', and below that two to the right of 'Females'. The two boxes on the left side are for Height. The two boxes on the right side are for Weight (which is not accurate, the 'weight' value should be called 'build'). The normal average value for both Height and Weight is 1.00 (per the game. This appears to translate into approximately 6 feet or 2 meters for Height). The maximum value that can be entered in any of these boxes is 2.00. The minimum number is 0.5. If you edit these values keep in mind that they need to stay in proportion to one another. If you make a race's Height 1.80 and their Weight 0.70 then that race will be so skinny the game cannot function properly (yes I tested this and it slows the PC down to unplayable). Conversely, if you make the Height 0.70 and the Weight 1.80 then the PC will appear stocky and the head, hair, and body meshes will not appear right (they will look squished). You can do some adjusting, but try not to go to extremes or a race will not appear correctly (and the game may not function). Another thing to keep in mind is not to make a race too tall. Any value over 1.40 or so for Height could cause problems for a race (they just don't 'fit' well in most interiors) and could cause problems. Don't be afraid to make adjustments, just be aware of what you do and how it may affect the game.

Playable. If checked, the race will be available in the start menu as a playable race.

Beast Race. If checked, the race is a 'beast' race that does not have normal biped leg movement (i.e. they walk like Argonians or Khajit, whose legs bend backwards when they walk) and will use the appropriate animations.

Ok. Click this button to close the Race Window. This does not save anything you have done, you must use the Save button on the Toolbar, or click Save under the [File](#) menu.

The Skill Window

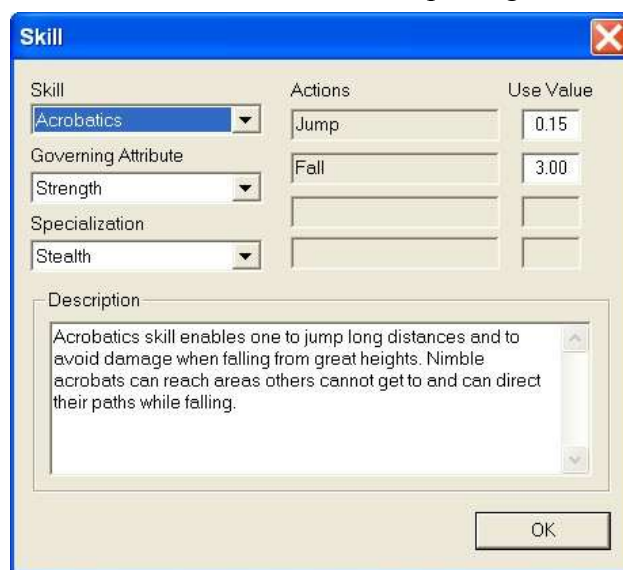
This brings up a small window that shows a drop down box listing all skills, what actions they affect in the game, a numerical value associated with an action to increase that skill, and an attribute associated with that skill. The only changes you can make to skills are the attributes associated with that skill, and the value to determine the increase of a skill.

Skill. This box shows the current selected skill.

Governing Attribute. This box shows the currently selected attribute associated with a skill. You can change this.

Specialization. This shows whether this skill is a Combat, Magic, or Stealth category skill.

Actions. You cannot add or delete these, however to the right of each action is a box showing the value added when using this action to increase this skill. In general terms, a value of 1.00 can be thought of as 'normal'. Increase this number and you progress faster in that skill when using that action. Lower this number and you progress slower. I do not suggest you make any changes to these values without some caution and much needed prior



thinking. Most of the skills are pretty well balanced as is (hey, I said most, not all. I know a few could use some minor tweaking... that's up to you).

Description. The description for the skill.

The Class Window

This window is used to create all classes within the game. All NPCs and the PC must have a class assigned to them, even if they are the only one with that class.

ID. This is the unique ID of the class used by the game.

New. Click this button to create a new class.

Delete. Click this button to delete the class shown in the ID box. A 'D' will appear next to the name.

Name. This is the name of the class that is used ingame and seen by the PC.

Primary Attributes. These are the two Attributes considered to be primary for the class by the game.

Specialization. The class specializes in *Combat*, *Magic*, or *Stealth* skills.

Note: The PC will increase skills that belong to this specialization faster than the other two. So choose wisely grasshopper.

Playable. Check this box if you want the class available at the start of the game for the PC.

Major Skills. Click each drop down menu and choose a skill that will be a major skill for the class.

Minor Skills. Click each drop down menu and choose a skill that will be a minor skill for the class.

Description. Enter a description of the class in this text box. This will appear ingame when the mouse pointer is held over the class name as a message box.

Auto-Calc Buys/Sells. On the NPC window, if the auto-calc box is checked, the NPC will Buy/Sell according to boxes checked in this section for their class.

Auto-Calc Other. As per Auto-calc Buys/Sells, but for *Training*, *Spellmaking*, *Enchanting*, and *Repair*.

The Birthsign Window

[Table of Contents](#)

[Table of Contents](#)

This window allows you to modify Birthsigns. Birthsigns give spells, powers, and abilities to a PC. Many consider the birthsigns as they are in Morrowind too powerful. If you wish to modify or create new image files for a birthsign, you should be able to use almost any decent graphics program. Most commonly used and/or mentioned in forums I've seen are Paint Shop Pro and Adobe Photoshop.

ID. This is the ID, but it is not displayed ingame.

New. This allows you to create and name a new birthsign.

Rename. This will change the ID of the current birthsign (now here's a button that could have been used elsewhere).

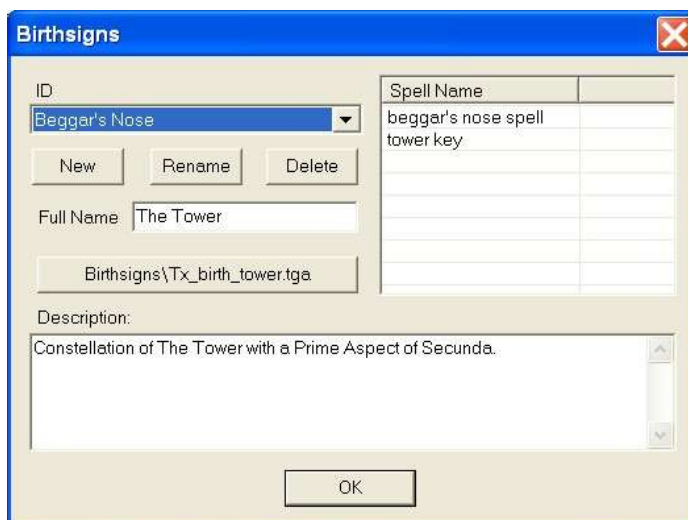
Delete. This will place a 'D' next to the current birthsign and it will be deleted when you next hit the save button.

Full Name. This is the name by which the birthsign is known and will be displayed ingame.

'Picture File' button. Click on it and you can reassign the picture file associated with the birthsign. This is the picture shown in the Birthsign Menu where you choose your birthsign.

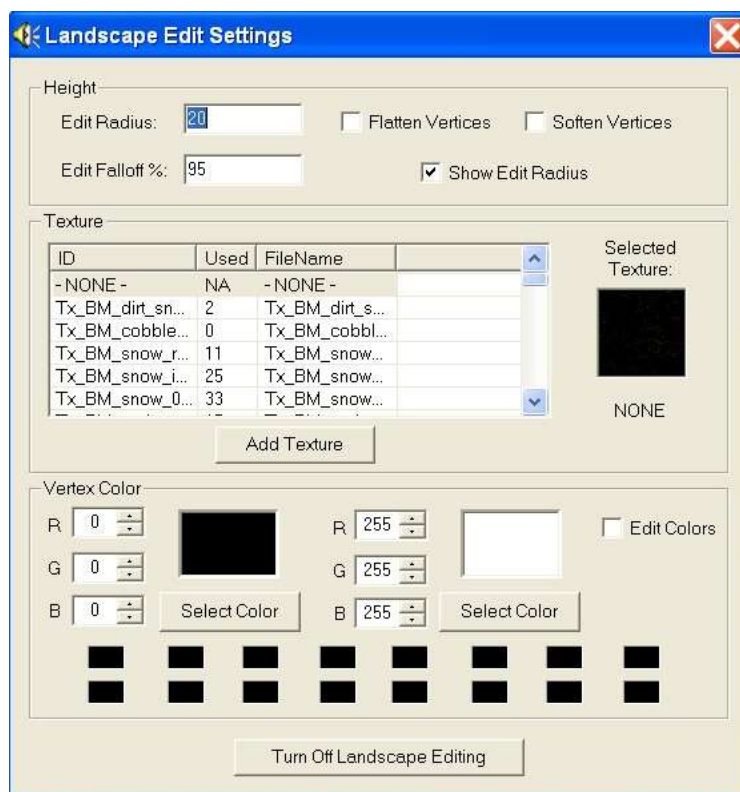
Spell Name. Here you may drag and drop any spell from the Spellmaking tab in the Object Window to associate it with a birthsign. Highlight a spell and push the 'delete' key to remove a spell.

Description. Here you may enter text to describe a birthsign.



The Landscaping Edit Window

Editing of landscape can be a major undertaking. It also will use most of your system resources to do (it shuts off many functions within the CS when open). Once you've gotten the hang of it, it usually takes no time at all to create or modify minor areas, something within one cell. If changing the landscape over multiple cells it might take a little longer (or maybe a lot). I have also noticed that making changes to the landscape increases the size of your plug-in quite a bit. Keep in mind that there are a lot of tutorials out there on the Internet, I'm not going to tell you how to create anything specific, just on how to accomplish changes with the Landscape Editor. First, some general information to know about the landscape within Morrowind.



- First and foremost, landscaping will add a lot of size to your mod. Creating one single new cell listed will add almost 80 kb to your plug-in. Start adding textures, plants, trees, actors, and other objects, and your plug-in's size will grow phenomenally. It's not the amount of change to the landscape done, it's the fact that change is present.
- A cell is composed of 8192 x 8192 "units". Each unit is 0.56 inches or 1.42 cm. Thus a cell is 4620 inches (385 feet) long and wide, or 11,633 cm (116.33 meters) long and wide.
- In wireframe mode the little triangles you see are the polygons that make up the landscape mesh. Each side of a polygon is referred to as a vertex. Each vertex is 128 units.
- You cannot modify the landscape horizontally, only vertically (up and down). This vertical plane is referred to as the z-axis.
- Water is set to a coordinate of zero (0) on the z-axis, you cannot change this fact (I know it sucks, just learn to deal with it). You cannot create waterfalls or place water above this. There are waterfall objects you can use with cells, but you cannot start water at one level and have it 'flow' to another lower level, period.
- The lowest point in the landscape is -2000 for a default cell. You can lower this quite a bit but not sure what the lowest point possible is. Similarly, not sure what the highest possible point is. The Volcano housing our favorite evil menace is only about 15,000 units in height (and it's still a bitch to climb). I've been able to lower the landscape to -34,000 units, and build mountains as high as 41,000 units.

- **Buffer Cells.** The CS will by default display or create two buffer cells around any cell you are working on in the Render Window. If you zoom outward so that multiple cells are visible you can see this (push the 'b' key to toggle cell borders). If you uncheck the **Allow Render Window Cell Loads** box in the [Preferences](#) window the CS will load any cell you click on in the Cell View Window, and render those around it, but you will only be able to modify the cell you clicked on, nor will you be able to scroll to adjacent cells. You will have to click on them in the Cell View Window to load them. If you click on a cell or scroll to a cell that does not have a cell adjacent to it, the CS will create cells named 'Wilderness' automatically. If you look you will find that the entire island of Vvardenfell and Solstheim both have a 2 cell buffer of cells called 'Wilderness' around both. Wilderness cells only exist as long as you are viewing them. If you make any changes within such cells however, they become permanent within your plug-in. If unintentional, you just dirtied your mod up. Remember, landscaping increases the size of your plug-in a lot, so unnecessary changes to landscape can really increase the size of your plug-in that you don't want. Have a care with Wilderness cells.
- **Movement within the Render Window.** You can use the Mouse to move about within the Render Window, or you can use the arrow keys for quick movement. Remember you can adjust the amount of the movement values by adjusting them within the [Preferences](#) window.
- **Remember to assign a Region.** If you create new landscaped cells you must create and/or assign the cells to a [Region](#). You most likely will also have to modify or add [Weather](#), sleep options, and [Sounds](#).

Mess with the Landscape Editor in a wilderness cell for a while before you decide to do some serious work with it if you haven't used it before (just don't save it). This will give you some experience and save some frustration when working on your plug-in. Let's look at the options within the Landscape Editor.

Height. These options change how you affect the landscape you are editing. All modifications to the landscape are done by clicking on a piece of landscape and 'dragging' it either up or down, depending on whether you push you mouse up or down. Only those vertices within the Edit Radius are affected when dragging the landscape up and down. The landscape will modify in multiples of 1 unit unless altered by changing the **Land Sensitivity Multiplier** under [Preferences](#).

Edit Radius. This number is the size of the radius that modifies the landscape. This is displayed as a red circle in the Render window. For large modifications use a number as big as 30 (largest possible) to produce deep bowl like depressions, or large rounded hilltops. Use a number as small as 1 for single simple modifications. Small numbers produce a more conical effect when the landscape is stretched up or down.

Edit Falloff %. This number is how sheer or steep the sides are of landscape you modify. The lower the number the less steep of the edges of the land you modify.

Flatten Vertices. Checking this box causes all land you click on to flatten to the same height within the Edit Radius. If you click and drag with this box checked you can effectively flatten mountains or level a large area of land quickly.

Soften Vertices. Checking this box causes all land you click on to become less steep. With each click all points of each vertex will adjust to a common average level on the z-axis, or simply put the landscape will attempt to become more flat. But this is a

slower adjustment than Flatten Vertices. You can click and drag the mouse to soften a large area just like with Flatten Vertices.

Show Edit Radius. Checking this box shows displays a red circle showing the landscape vertices you are able to modify. It's on by default. A value of 30 is the size of an entire cell. You affect all vertices within this radius, regardless of whether or not the radius is visible.

Texture. Textures are what you use to 'paint' the landscape. It is not entirely the color of the land, but also what exists on that vertex; grass, mud, ash, swamp scum, etc (though since this does add color you can think of it as 'coloring' the landscape). You can only paint one 'square' at a time. To paint a larger area, r-click and drag the mouse over a larger area, then release. This paints a large area, however, if you click on undo only one square will be undone. That's a lot of potential undo clicking. It's usually safer to paint single squares or smaller areas. The landscape editor radius has no bearing on painting textures.

***Note:** The CS Help file states it is not a good idea to let more than two textures meet. The game can combine two textures to produce a blended transition area between two textures, but does not do this well with more. In essence, don't let more than two textures touch at a time. If you move some large rocks and statics you will find areas underneath where three textures meet. Apparently this is how Bethesda chose to cover these 'ugly' areas*

Texture list box. This displays the textures available for use by ID and name. Merely highlight a texture, and then l-click on any square within the Render Window to paint it.

Add Texture. Click this to add a texture to the database. By default this brings up the CS CD if this option is enabled, else you will have to navigate to the texture file.

Selected Texture. This will display a sample of the texture. Let's you find exactly what you want.

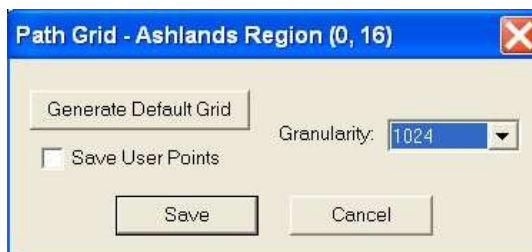
Vertex Colors. This allows you to paint colors on the landscape, usually to add shadows and/or highlight (but usually for shadows). You must check the box 'Edit Colors' to paint with vertex colors, otherwise you will be modifying the landscape height. You can paint with two colors at once; l-click on one color and r-click on another color. You can now paint a different color with each button. There are two Add Color buttons, one for the left and right mouse buttons. To the left of this button are the R, G, and B boxes; you can define a color by entering numerical values if you know them. If you click the Add Color button it brings up a standard color selection palette for you to pick a color. Above the button is a display box to show you the color chosen. At the bottom of this section are sixteen boxes to store a color in for repeated use so you can switch colors. Just click on one of these and store that color in the respective left or right mouse button.

Turn Off Landscape Editing. This button will turn off the landscape editing engine.

***Note: Placing Objects:** Most objects for terrain can be found under the [Statics](#) tab in [The Objects Window](#). Actually, any object can be placed on landscape. Just click and drag the item into [The Render Window](#). To learn about objects and how to deal with them refer to [The Object Window](#).*

The Edit Cell Path Grid Window

To use this function you must first have a cell loaded and visible in the Render Window. Click the button and the small Edit Cell Path Grid Window will appear. Nodes appear blue and the paths between nodes appear as parallel yellow lines. If a cell already has a path grid it will be visible in the Render



Window. If none is present you can generate one or just begin placing nodes by r-clicking, which will place a blue node in the Render Window. To create a path between nodes select a node by clicking on it, then hold down the 'control' key and click on a different node, a yellow path will be created between these two nodes. Nodes are objects and can be treated as such for movement purposes.

- NPCs and Creatures will use the same Path Grid.
- Both will use the Path Grid most of the time, they can leave the Path Grid when circumstances provide a reason.
- The Path Grid cannot extend beyond the cell it is created in. However, if a creature or NPC wanders to the limit of a grid, they can leave it to enter onto another grid at the edge of the cell they are entering if the next grid path or a node is relatively close. This doesn't mean they will, only that they can.
- NPCs can wander about if they have the WanderAI, regardless of whether or not a Path Grid is present. If present they will use it. If not present, they move about according to some random coding within the game.
- A Path Grid can be moved upwards on the z-plane for aerial creatures. They will move (fly) along this grid path.
- Do not generate a grid path if one already exists in a cell, this will screw things up (by creating a mess).

***Note:** If you want an NPC or creature to wander along a certain path, simply create the Path Grid along that route. If you want them to move along that route according to a certain pattern you will have to script that and attach it to the NPC or creature.*

Generate Default Grid. This button will, when clicked, generate a default grid based upon the Granularity chosen.

Granularity. This drop down menu allows you to choose a number to generate a path grid.

The numbers (I believe) are the distance between path nodes in units. This means the smaller the number, the more nodes are generated. A granularity of 128 took almost three minutes to create a grid on my average system. It also created so many nodes I almost couldn't see any landscape beneath it.

Save User Points. Check this box before you hit the Save button to save changes you've made if you add or delete to the Path Grid. If this isn't checked only changes made to prior exiting nodes and paths will be saved.

Save. Click this to save the modifications you have made. This will save the changes in the Render Window, but not to your plug-in. Click the Save button on the CS toolbar to save it to your mod.

Note: Be careful where you put a path node. There is a reported issue where if you add a node and it ends up occupying the same space as a creature (or possibly another object), it will disappear and reappear at grid 0, 0, 0. You probably don't need a stack of nodes there, do you.

The Factions Window

This window can be confusing at first, but doesn't take long to learn. A faction can be a guild, a 'house' or clan, a tribe, a culture, or a religion (the Nereverine is a 'faction' of only one rank). You can even make a faction that isn't a faction (of sorts). Use your imagination. I personally feel this to be a very powerful aspect of the game that isn't used to its full potential by most modders out there (excepting a very few, and

bless them for it). Keep in mind this window only lists the defining data for factions; the promotion, inclusion, or exclusion, of the PC is done with Dialog and/or Scripting.

ID. The unique name ID of the faction.

Name. The name by which the faction is known ingame.

New. Click this to create a new faction.

Delete. Click this to delete the faction shown in ID. A 'D' will appear next to the ID in the ID box. The faction will not actually be deleted until you Save your plug-in.

Hidden From PC. If checked, the faction will not be displayed or known to the PC ingame via the stat window.

Requirements. This section shows what it takes to join a faction and the ranks, along with minimum stats to be that rank.

Attributes. Two drop down menus. You must choose which two Attributes are preferred by this faction.

Favored Skills. This box lists all skills in the game. You choose which ones are favored by this faction (must pick at least one). To pick just scroll and click on a skill, it will become highlighted and move to the top of the list, in the order of being chosen. Click

a highlighted skill and it will be removed as a favored skill. The order makes no difference.

Ranks. Ranks must appear in the columned box in descending order. This means the topmost listed rank is the lowest level in the faction. The second listed is the next highest in the faction, and so on. When you enter values to define the minimum requirements for a rank, they cannot be lower than any rank listed as lower in level within the faction. Ex: A Layman in the Imperial Cult has an Attribute1 requirement of 30, a Novice (next higher) cannot have a requirement for Attribute1 of less than 30, it must be equal or higher. This applies to all the requirement columns, for all ranks.

Rank name. The name of each rank can be changed or entered by l-clicking on a line under the Rank column so that it is highlighted, then l-clicking again on it. You may now enter a name for a rank.

Requirement boxes. To the right of the two drop menus for choosing preferred attributes are five boxes. These correspond directly to the five columns listed to the right of each rank name listed below. You cannot input a value next to a rank name, you enter the value instead in one of these boxes and it appears next to a highlighted rank name below. The requirement values are the minimum level for advancement to that rank.

Attribute1. First box. Minimum attribute level for Attribute1.

Attribute2. Second box. Minimum attribute level for Attribute2.

Primary Skill. Third box. At least one favored skill must meet this minimum level.

Favored Skills. Fourth box. All favored skills must meet this minimum level.

Faction Reaction. Fifth box. This is the bonus applied to the reaction of everyone from the same faction for this rank.

Reactions. This is a list of all other factions and a reaction bonus/penalty to anyone of that faction when they interact with the PC. The left column shows a faction, and the right column shows the reaction bonus/penalty. To change the faction name l-click to highlight the name, then l-click it again, and then go to the drop down menu to the right and click on it. Now choose the faction you want in that spot. Enter the reaction value (either positive or negative) in the box below the drop menu. If you wish to add or delete a faction, r-click over a faction name or a blank row in the first column, choose New or Delete. If you pick new, the current faction Name will appear, ignore this and just choose a name from the drop down menu to the right, and then enter a reaction value.

The Dialog Window

While it may seem confusing when first approached, in concept it really isn't that difficult. But it can be very, very easy to screw up a quest or how an actor functions, and not even realize it, if you don't pay good attention to what you do within the dialog window. I recommend having at least a basic idea how scripting works before you dive into dialog. Dialog works pretty much the same way, but is 'viewed' entirely different, and provided and accessed in a simple to use window (instead of writing everything out, you just click the drop down menus to choose IDs, functions, and conditions). Once you learn how to do one of them (dialog or scripting), the other becomes easier to understand. Some people find scripting easier to understand, others think dialog is easy, but struggle with scripts. I myself struggled horribly for the longest time trying to

understand dialog, but once it clicked, I had no trouble with it. There are some general rules and conventions you *must* understand before you mess with dialog.

- Dialog is one big database table, even though in the window it is divided into sections.
- When you make any change to dialog, *it is immediately changed in the database*. There is no 'save' button to make it permanent, it automatically is. This means you cannot hit the Undo button to undo a mistake you made! However, while the dialog database is changed, no changes are made to your mod until you hit the Save button on the Toolbar. Your options are then to either close the CS to prevent a mistake from being permanent, OR if already saved, clean the mistake from your mod (see [Cleaning the Unclean](#)). This is very important to remember! You do not have an Undo button to jab if you make a mistake!
- Never, ever, create a new dialog response at the very top or very bottom of a section of dialog that's already been created (by Bethesda or another modder). I won't explain why, just trust me this *will* screw things up for someone somewhere. Always put newly created responses somewhere in the middle.
- **Never delete a dialog topic from a master file!** If you want to prevent a response from taking place, create a new response more appropriate to your mod, and set conditions and functions so it is used instead. Deleting dialog responses from the Morrowind master file can lead to removing the ability of the PC to initiate dialog in a game! If the game is saved by a player, this problem can be saved along with it, and removing your mod will NOT fix the problem. This will not endear you, or your mod, to anyone.
- Dialog responses each have a unique ID assigned to it. That ID is linked by the CS to the ID of the dialog response both above, and below it in the section it is listed in. This is why when you create a new dialog response, the new response, and the responses above and below it will display an asterisk (*) showing they've been changed. This is normal and ok. There is no need to worry about 'cleaning' this from your mod. However, this means you have to pay attention when looking through dialog to see what unintentional changes need to be cleaned out.
- You cannot add a new dialog response when the dialog window is filtered. There is a very good reason for this.
- It is better to create a New dialog response, than to Copy one and alter conditions and functions for a response in a Master file (i.e. MW, TB, or BM). The reason is that if you right-click over a response, and choose copy, the new copy is placed *above* the old copy in the response list. The new copy will have a new ID, the old keeps the one it had. The problem is when you don't pay attention, and now alter the *bottom* one, instead of the *upper* one, which has the new ID. Thus your altered old response will be used instead of the new one (i.e. the game uses the wrong response). This does not apply to responses you create for Topics, as long as the Topic is unique and not used by anyone else. It DOES apply to greetings, no matter if yours or not. Did this make sense? No? Then just trust me on this and create new instead of copying.
- The PC must know a topic before they can be given responses by an actor for it. A topic can be introduced to the PC if typed into a dialog response, or by the function AddTopic (via script, or the Results box in a dialog response).

Note: *It's a good idea to always use the AddTopic function when introducing a new Topic. If someone creates a Topic that includes yours in it, your Topic may not appear for the PC. But if you use AddTopic, while it may not be accessible by clicking on a hyperlink in the response text, your topic will still appear in the Topic list on the right side during dialog.*

- When you create a new Topic or Journal, the game will create an 'EMPTY' response, and place it at the top of the respective section. This is normal, and ok. The next time you load your mod the CS will give you a warning about a 'journal becoming a topic' or some such. Ignore it. Once loaded the CS will clear that out for you when you save.
- When you create new Journal IDs, use the method Bethesda used (ex: A1_nameID). Else wise, the game may think you're creating a topic instead, and your new journal may appear under the Topics tab (remember how I said dialog is really one big database?), and be treated as such.
- Journal responses are referred to by index number. They only appear in the player's journal.
- If a script uses the MessageBox function, and the dialog window is open ingame, the text from the MessageBox will appear in the dialog window as white text.
- Many responses in the dialog database are generic. The moment you create an NPC, they will have responses to certain topics by default. Usually due to race, class, region, city, or faction. They also will have several standard default general topics (such as *Morrowind Lore*, *Little Advice*, *Little Secret*, etc). These default responses can be prevented from being used by the use of variables (*NoLore*, etc. see below).
- You can use the arrow keys to move within the response lists. Using the Up and Down arrow keys you can move within a list of dialog responses. Using the Left and Right arrow keys actually *moves* a response up or down within a response list. Be careful when doing this, it's very easy to move a response out of proper sequence (as I have found out), and can dirty up a mod.

How dialog works.

Ok, let's get to the meat of the matter. When the PC initiates dialog with an actor, the game first goes to the database and looks at Greetings 0. In the list of responses, it will look at the very first dialog response listed at the top of this section. It will look at all the *speaker conditions* for that dialog response. If the actor meets all the speaker conditions set for that response (meaning they all return true), then the response is displayed ingame for the player to see and read. If any of the speaker conditions return false, then the whole dialog response is false, and the game skips to the dialog response next in the list (directly below). The game now checks the speaker conditions for this response. If all return true, then it is displayed ingame for the player, and if not, then the game skips it and goes to the next response below it. This continues until the end of the response list is reached for Greeting 0 section. If this happens, then the game goes to Greetings 1 section, and the process repeats itself until a response returns 'true'. If no responses in Greetings 1 section return true, then the game goes to Greetings 2, and so on. The game will continue in this manner, checking the entire response list of each Greetings section, from top to bottom, until it finds a response that returns true, which gets displayed ingame for the player. If none return true (for some reason), nothing gets displayed.

Ok, so the player has been given a greeting. Whoopee. Now what? Well, as you have seen topics will appear on the right side of the ingame dialog menu. The player can click one of these.

Let's say he does. The game will now go into the database and find that topic. When it does, it will go to the top of the response list for that topic, and checks the speaker conditions of the very first dialog response listed. If all return true, that response is displayed. If any return false, the next dialog in the list is checked. This works exactly the same as Greetings do, except the game will only check for responses from the appropriate topic. If no responses return true, nothing gets displayed. As you have seen when playing (you have played at least once, right?), the player can click on any topic, either from the right hand menu list, or from within a response, to get a response for that topic. Topics are always displayed **bold** and blue ingame (if the PC knows the topic), within a dialog response. This is because it is a hyperlink.

All topics are hyperlinked to the set of responses for it that can be displayed ingame. But as mentioned, the PC must know the topic. The PC can be exposed to the topic (and thus learn it) either by using the topic as part of a response, or by using the AddTopic function.

An example of use in a response; assume the topic is: boating

In a greeting or as a response to another topic an NPC responds, "I like the feel of the sea about me, don't you enjoy boating?"

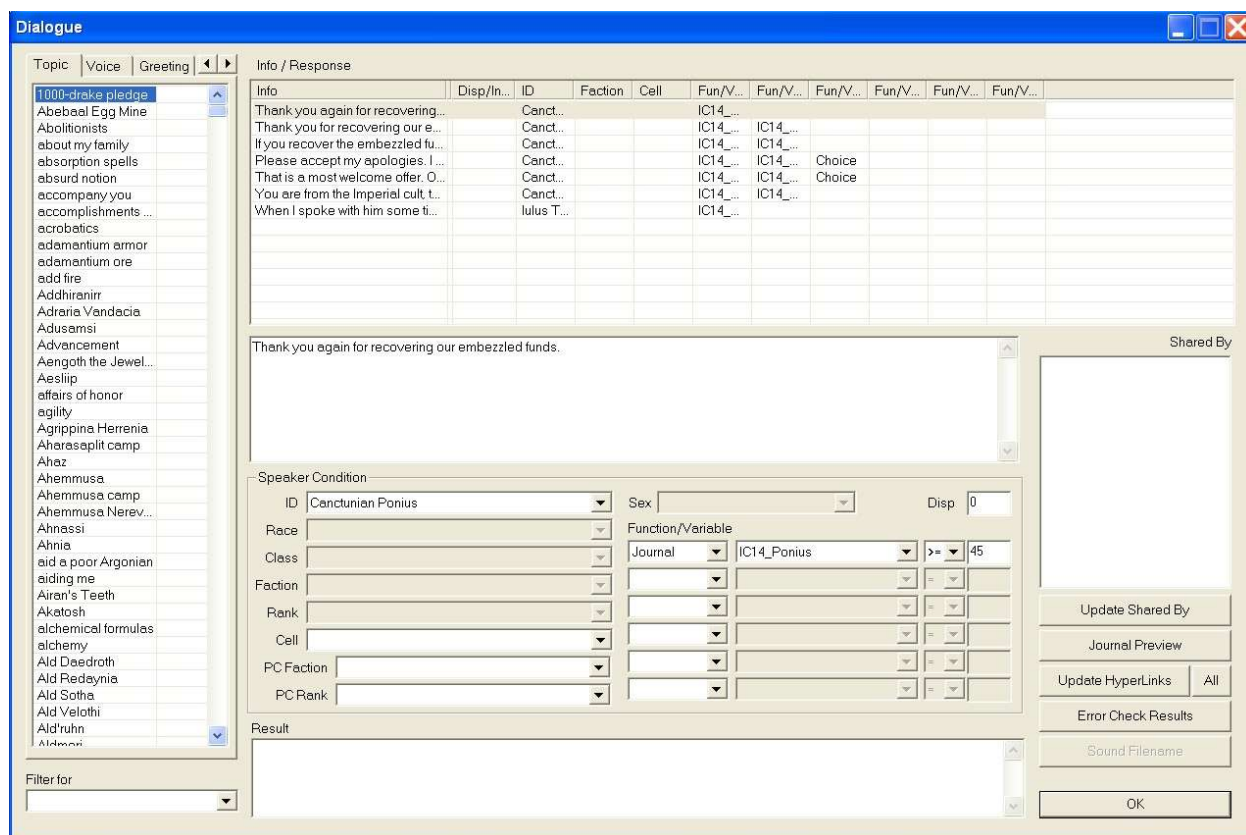
In this example, the NPC used the word 'boating', which also is a topic. Now the PC has been exposed to the topic. If the any of the responses for 'boating' would return true for this NPC and the PC, then it would appear blue (i.e. hyperlinked), and the PC would 'know' it. From this point on, any actor that has responses for boating may speak about them with the PC, because they know the topic (assuming of course, the speaker conditions return true).

Now let's say the topic boating does not appear blue, for some reason none of the responses the NPC has return as true. The PC will still know the topic, even though they can't get a response from this NPC. They may now get a response for boating from other NPCs, just not this one (though later they could come back and maybe get a response).

Speaker Conditions.

A speaker condition within dialog can be almost any aspect or stat of an NPC or PC. A condition could be the actor's race, class, faction, rank, cell they are in, item they have in inventory, journal index, etc. Almost anything. Whether or not a condition returns true is dependent upon the *parameters* of the condition. You could set a response to only male characters, thus only male characters would ever see that response. If you set conditions so that only a male dunmer PC of House Telvanni, while in Balmora, the player would only ever see this response if their PC was a male dunmer, currently in good standing, and belonging, to House Telvanni, while they were in a Balmora cell. Note however, this response would be given by ANY NPC within Balmora. If you want only certain NPCs to say this, you would set further conditions; maybe the NPC has to be female, in the Mages Guild. This would limit it to only six NPCs (in the Balmora Mages Guild). If that's too many still, then assign the response to a singular ID. If you do this, parameters for the NPC become grayed out (as they're unnecessary). Now, only that assigned NPC can give that dialog response.

Let's look at the Dialog Window, and see what we can do here.



Dialog Tabs. The different sections of dialog are listed as tabs on the top left hand side. These tabs are **Topic**, **Voice**, **Greetings**, **Persuasion**, and **Journal**. Greetings are further divided into Greetings 0, Greetings 1, Greetings 2, etc. Each of these separate Greetings have responses in them. The Topic tab is divided up into topics. The topics each contain one or more responses. Voice and Persuasion only have responses. Journal has journal IDs within it, and each ID has a set of indexes (identified by a number).

Info/Responses. Here you will find all responses listed. The top box is the *Info*. Scrolling from left to right, you can at a glance see the text of the response, then any conditions that apply to each one. The bottom smaller box is the Response. This has a maximum character limit of 255.

Speaker Conditions. Here is where you set conditions for when a particular response will appear ingame, and with what actors they will appear. Highlight a response, and you can see the parameters for that response in this area.

ID. You can set a response to a specific actor. The actor can be an NPC or a Creature.

Race. This sets the speaker's race as a condition.

Class. This sets the speaker's class as a condition.

Faction. This sets the speaker's faction as a condition. If you choose this you must also set the rank in the faction.

Rank. Set's the rank in the speaker's faction

Cell. You can set the cell for the speaker's condition (the response will only appear if the speaker is in this cell).

PC Faction. This response will only appear if the PC is in the listed faction.

PC Rank. If you set the PC Faction condition, you must assign the PC's rank.

Sex. This sets the speaker's gender as a condition.

Disposition. This will set the speaker's current disposition as a condition. This will return true if the current disposition is equal to or greater than the numerical value entered here. When zero, it always returns true.

Function/Variable. From these six lines of drop menus you can assign an array of functions and variables as conditions. The first drop menu is the type of condition. The second drop menu is an ID, variable, or a further defining of a function. The third drop menu is the type of check you wish to do for the condition. The fourth box is a numerical value to base your function check on. Let's look at each one.

1st drop menu. Click the drop menu and you will see a list of function and variable to choose as the condition you wish to check for a response. They are as follows:

Function, Global, Local, Journal, Item, Dead, Not ID, Not Faction, Not Class, Not Race, Not Cell, Not Local. Choose one of these, and you then click on the next drop menu to further define the condition. Let's look at these.

2nd drop menu. I will list each Function/Variable, and the condition that appears in this drop menu for it.

Function.

Alarm. This is the base alarm level for the actor (not if the actor is alarmed).

Alarmed. This is whether the actor is alarmed. The value set in the last box should either be a 1 for alarmed, or a 0 for not alarmed.

Attacked. This is whether the PC has ever attacked the actor. The value set in the last box should either be a 1 if ever attacked, or a 0 for not attacked.

Choice. If you set up a response to have one of more choices to be clicked on, this is the function to identify each choice to the game. That means, if you set a response to have three answers, 1, 2, and 3. You then create each answer as a response, and use this function to 'label' them as 1, 2, or 3. I'll explain this better farther below.

Creature Target. This is used to identify Voice responses if the actor is in combat. Set this to 1 to test if a target is a creature, set this to 2 to test for a target that is a werewolf (with Bloodmoon).

Detected. This means if the actor has detected the player (ex: the player tries to sneak up behind an actor to pickpocket them. Upon hitting the spacebar to pickpocket the actor, they are detected, and the dialog menu opens instead of the actor's inventory). The value set in the last box should be a 1.

Faction Rank Difference. This is a check for the difference in rank between the actor and the PC. Both must be in the same faction. The value set in the last box should be the difference in value. A positive number means the actor is lower in rank, a negative value means the actor is higher in rank.

***Note:** If the PC is not in the faction, a value of -1 is returned for the purpose of this function.*

Fight. This is the base Fight value. The value set in the last box is the Fight value or lower for which you want this response to display.

Flee. Same as Fight, except applies to the Flee value.

Friend Hit. If the actor is a 'friend', you can test the amount of times hit by the PC (could be a follower, NPC escorting the PC, companion, etc). Mostly used under the Voices tab; a different vocal response is set to each value for each race. Values to be used in the last box are listed below.

- 0 = Never been hit by PC.
- 1 = Hit by PC one time.
- 2 = Hit by PC two times.
- 3 = Hit by PC three times.
- 4 = Hit by PC four or more times.

Health Percent. The health level as a percentage you want a response to display. The value in the last box must be between 1 and 100.

Hello. Same as Fight, except applies to the Hello value. Most NPCs have a value between 0 and 30. For most creatures it is 0.

Level. Not sure why it's needed, but this is the actor's level. Set the value to the desired level or lower to test for.

PC [skill]. All skills the PC has are listed in the drop menu. Set the value to the desired level or lower to test for.

PC [attribute]. All attributes of the PC are listed. Set the value to the desired level or lower to test for.

PC Blight Disease. If the PC has a blight disease. Set the value to a 1 for a blight disease, or 0 to not have a blight disease.

PC Clothing Modifier. The clothing modifier is the total value of all items equipped by the PC (armor, clothes, amulets, rings). Set the value in the last box to the monetary value or lower you want the response to be displayed. A value of 0 means naked.

PC Common Disease. If the PC has a common disease. Set the value to a 1 for a common disease, or 0 to not have a common disease.

PC Corpus. If the PC has corpus. Set the value to a 1 for corpus, or 0 to not have corpus.

PC Crime Level. This is the monetary value that has been placed on the PC's head for crimes they have committed (i.e. bounty).

PC Expelled. Test to see if the PC is currently expelled from the actor's faction. Set value in last box to 1 for true, or 0 for false.

PC Fatigue. This is the current fatigue level of the PC.

PC Health. This is the current health level of the PC.

PC Health Percent. This is the current health level of the PC expressed as a percentage (i.e. from 1 to 100).

PC Magicka. This is the current magicka level of the PC.

PC Reputation. This is the current reputation level of the PC.

PC Sex. This is the gender of the PC. a 1 will test for female, a 0 for male.

PC Vampire. This will test if the PC is a vampire. A 1 is true, a 0 is false.

PC Werewolf Kills. This will test how many kills the PC has made as a werewolf.

Rank Requirement. This will test if the PC has the requisite skill and stat levels to be promoted in the actor's faction. Set the value to test for according to below:

- 0 = You do not have enough Faction Reputation and do not meet the skill requirements.
- 1 = You meet the skill requirements, but do not have the Faction Reputation.
- 2 = You have the Faction Reputation, but do not meet the skill requirements.
- 3 = You qualify.

Reaction High. This function tests the faction reaction of the actor's faction against the PC's factions, returning a value equal to whichever is the highest. Set the value you want to test for in the last box.

Reaction Low. Same as above, but tests for the lowest.

Reputation. This is the actor's reputation level. Set the value in the last box to the level or lower you wish to test for.

Same Faction. Set the value in the last box to a 1 to test if the PC is the same faction, to a 0 if not.

Same Race. Set the value in the last box to a 1 to test if the PC is the same race, to a 0 if not.

Same Sex. Set the value in the last box to a 1 to test if the PC is the same gender, to a 0 if not.

Should Attack. Set the value in the last box to 1 to test if the actor wants to fight you.

Talked to PC. Set this value to a 1 to test if the PC has talked to the actor before, to a 0 if not.

***Note:** The game only keeps track if an actor has talked to the PC within the last 72 hours (game time). After that, it 'forgets'. If you want an actor to remember that they talked to the PC beyond 72 hours. Declare a variable in a script attached to the actor, no need to even set it to any numerical value, just declare it. In the dialog response set it to a value of 0 for the first time they speak to the PC (as a speaker condition), and change it to a 1 in the Results box in that same response. The actor will never offer that response again.*

Weather. You can set a response according to the current weather if in an exterior. Values that can be tested for are listed below.

- 0 = Clear
- 1 = Cloudy
- 2 = Foggy
- 3 = Overcast
- 4 = Rain
- 5 = Thunder
- 6 = Ash
- 7 = Blight
- (8 = Snow?)

(9 = Blizzard?)

Werewolf. Set the last box to a value of 1 to test if the actor is in werewolf form. It's used to prevent werewolf form actors from having dialog with the PC.

Global. You can set a variable that is declared in a global script, or set from the Global Variable window. Open the drop menu to see all global variables listed that you can choose from. Set the value in the last box you wish to test against.

Local. You can set a variable that is declared in a local script as a condition. Be careful here, if you use the same named variable in more than one script, you could be setting yourself up for failure (i.e. DoOnce, NoLore, etc). Open the drop menu and choose the variable, and then set the value in the last box. The local variable *must* be declared in the script for the NPC you test against.

Journal. You can choose a journal ID to test against. Open the drop menu and you will find a list of all Journal IDs. Set the index number you wish to test against in the last box.

Item. It sounds repetitive, but again, you can test against any item created. This will only check the inventory of the PC for the number of items that you specify. Thus, in the last box set the item count you wish to test against.

Dead. This will test if an actor is dead. Open the drop menu and choose the NPC or creature ID. Set the value in the last box to number of actors dead you wish to test for.

Not ID. This will test for the current actor you are in dialog with. Open the drop menu and choose the actor ID, then set the value in the last box to 1. What this means is any actor *except* the one you choose can display this response.

Not Faction. This will test for the current actor's faction you are in dialog with. Open the drop menu and choose a faction, then set the value in the last box to 1. What this means is no actor who belongs to the faction you choose will give this response.

Not Class. This will test for the current actor's class you are in dialog with. Open the drop menu and choose a class, then set the value in the last box to 1. What this means is no actor who belongs to the class you choose will give this response.

Not Race. This will test for the current actor's race you are in dialog with. Open the drop menu and choose a race, then set the value in the last box to 1. What this means is no actor who belongs to the race you choose will give this response.

Not Cell. This will test for the current actor's cell you are in dialog with. Open the drop menu and choose a cell ID, and then set the value in the last box to 1. What this means is no actor in the cell ID you choose will give this response.

Not Local. This will test for the presence of the variable you choose in a script attached to the actor you are in dialog with. Open the drop menu and choose a faction, then set the value in the box to 1. What this means is no actor that has this variable declared in an attached script will give this response.

3rd drop box. Open this box and set the 'test' you wish to make for the condition. This is either ==, <=, >=, <, >, or !=. This tells how the function/variable is to be tested when compared to the numerical value you place in the last box.

4th or last box. The last box is where you place the numerical value to test if the condition is true, or false. Some conditions only require a 1 or 0 in this box (true or false). Others require a larger number. The numerical value must be a whole number.

Shared by. This box on the mid right side will show what actors can give a response (when you click on a response). This makes it easy to tell if an NPC or creature will give this response or not (or maybe see if someone is listed who *shouldn't* give this response).

Update Shared by. If you change the speaker conditions for a response, then click this button, the Shared by box above will be updated.

Journal Preview. When you click and highlight a journal index, then click this, the CS will create the ingame journal and display the journal index text so you can see what it will look like. Obviously, only works with journal indexes.

Update Hyperlinks. When you click and highlight a response, then click this button, the CS will place tags around any word that it considers a topic. Thus if the word boating is a topic and used in a response, click this button and it will appear as <boating>.

All. This will Update Hyperlinks for every dialog response.

Error Check Results. Clicking this will force an error check of all dialog. This isn't the most trustworthy function of the dialog window, but it will catch simple mistakes you make, but only in syntax. It won't tell you that you just screwed up dialog for an NPC, but it will tell you if you misspelled a script function, or used incorrect script syntax. It won't tell you if you've set a variable to an incorrect value, or used the wrong function to do something, only that you used incorrect syntax. Once you click the button, pay close attention to each error warning that may come up, write it down if you can. It won't tell you exactly what you did wrong, but it will list the problem response, and give a 'script error' clue. This usually is enough to figure out which responses are problems, so you can find them and figure out what the problem is, and fix it. Vast majority of the time, I have found that the mistake is a misspelling of a function, or incorrectly stating syntax for a script function in the Results box. One problem the Error Check has is that sometimes it will generate an error prompt when there is no actual error. This is easily seen if you have Tribunal or Bloodmoon loaded. Some of the dialog from each will always generate errors, even though there is nothing wrong with them. What it boils down to is while using this function can be a big help, it's not perfect.

***Note:** Do NOT SAVE your mod after running an error check if you find errors! For some reason the CS will save the fact that you had errors as part of your mod. Sounds crazy, but it's true, and it will just add data to your file that has no use. Instead Save your mod, then run an error check. If you have errors, write them down, then exit and re-load your mod. Now safely fix your errors.*

Sound Filename. This is usually grayed out, and is only functional when working within the Voice section of dialog. Use this to assign a sound file to a response within the Voice section (i.e. when a response is to be given vocally ingame).

Results box. Think of the Results box as a miniature script window. In the Help file that comes with the TESCS, it states that single line script functions can be added here, but no if/endif sets can be used. This isn't entirely true. What is true is that script functions added here are not compiled by the CS. During play when this response is called, anything added here is compiled and executed at that time. If/Endif sets can be used

within the Results box, but what you have to keep in mind is that this box will only execute *once*. Thus this limits what you can do with the Results box. Normal scripting limitations also apply. Examples of common functions and commands used:

```
set some_variable to 1
SetFight, 100
player->ModCurrentHealth, 50
"lirielle stoine"->AddItem, "steel longsword", 1
Journal, "B3_ZainabBride", 25
Set companion to 1
StartCombat, player
StopScript,
ModDisposition, 5
AIFollow, 0, 0, 0, 0, 0
Goodbye
Choice, "Yes, 1, "No", 2
```

If you use the StartScript function, put it at the top of the Results box above any other functions. This will ensure your script always gets started. You'll note I put the last two in bold. This is because these two are unique to dialog, they can only be used in the Results box. Goodbye will force the player to click on the word, which will end all dialog with the current actor. Choice works exactly the same as MessageBox (strangely, you can still use the MessageBox function in the Results box, but you cannot have buttons. Only Choice will allow you to do multiple choice answers, and then not as buttons, but as hyperlinked answers). Unlike with MessageBox, with Choice you must number each possible 'answer'. You then set a speaker condition line to "Function" "Choice" "=" "[number]" in the response you want to appear for the answer the player chooses.

Note: *It is important to remember that a response to a choice must be listed above the original response that offered that choice!*

Special Variables. There are several special variables that are used that you can make use of also. To use these simply declare them in a script attached to the actor, then use the Not Local function as a condition with variables (except for Companion and StayOutside, use the Local function instead). There is no need to assign a numerical value to them. They are as follows.

NoLore. There is no need to assign a value to this variable. When declared in an actor's script, many default topics will not be used by the actor. In the CS the topics will still appear if you filter for an actor's ID, but they won't appear in the game.

NoHello. Used by Bethesda to prevent normal hello vocals to be used (by vampires) under the Voices tab.

NoFlee. Used by Bethesda to prevent normal flee vocals to be used (by vampires) under the Voices tab.

NoIdle. Used by Bethesda to prevent normal idle vocals to be used (by vampires) under the Voices tab.

NoIntruder. Used by Bethesda to prevent normal intruder vocals to be used (by vampires) under the Voices tab.

NoThief. Used by Bethesda to prevent normal vocals being used when the PC is detected stealing. And of course this includes vampires.

Companion. Used to allow companion sharing with NPCs. Declare as a short variable in a script, and have it set to 1 to allow companion sharing.

StayOutside. Used to set a companion to only follow you in exterior cells. Declare as a short variable in a script. When set to 1 the actor will not follow the PC through a door that enters an interior cell.

Topics. This tab holds all the topics that the PC can learn/know and click on to carry out conversations with an actor.

Voices. This tab holds all the vocalized responses that actors can 'say' ingame. These are responses you add a sound file to, such as when an actor gets hit, hits the PC, is standing around idle, first 'detects' the PC, etc. If you look through this list you will see that most responses are generic to a race, class, or faction. If you have Bloodmoon also loaded you will see that at the top responses for each section are for werewolves. These are for NPCs when in werewolf. Almost all creatures have standard sounds associated with them already from within the [Sound Generator](#) window, there is no need to add sounds for them here (though, technically, you could). To add a new vocal response, simply right click where you want the new response to appear. Type in whatever dialog you want displayed (if the player has this function on when playing), then click the Sound Filename button on the bottom right side of the dialog window. A window will open allowing you to navigate and select the mp3 file you wish to use. That simple.

Greetings. Greetings are a little different. There are 10 sections, or sublevels, to Greetings. They are labeled 0-9. The type of responses that go in each are specific, if you place a response in a sublevel different than what Bethesda put there, you could be breaking a dialog thread for an NPC. You *must* take care and think before you put any greeting response in a sublevel in order to avoid this. Types are listed below.

Table 4-3: Dialog Greeting Sublevels.

Sublevels	Type of responses that go there.
Greetings 0	Alarmed responses.
Greetings 1	Responses where it doesn't matter if the PC is a criminal, diseased, a vampire, or nude.
Greetings 2	PC is a vampire or nude responses.
Greetings 3	PC is a traitor to the Morag Tong responses.
Greetings 4	PC is a criminal or diseased responses.
Greetings 5	Most general quest responses.
Greetings 6	Faction based responses.
Greetings 7	Class based, endgame, and slave responses.
Greetings 8	Clothing based responses.
Greetings 9	Location based responses.

Note: Once the main quest is complete, no responses below Greetings 7 will occur. That's because everyone will respond with one of several from Greetings 7 once the main quest is

complete, thus any response listed below them will never occur. Something you need to be aware of, and keep in mind.

Persuasion. Under this tab you will find all forms of persuasion the PC can attempt to use on an actor. Note that most are generic. You will also find service refusals, and info refusals here. If a PC clicks on but does not qualify for a service an actor offers (whether merchant, travel, training, etc.) a service refusal will be the response. Info refusal is similar. If the PC clicks on a topic, but does not meet requirements (such as disposition level, correct class or faction, rank, etc.) they will get a standard info refusal. The exception to this is if a specific response is prepared instead, in which it would be used instead.

Journal. Here are listed all journal responses that can get added to the PC's ingame Journal. Each quest in the game, and certain situations, have an appropriate Journal ID. Each ID is of course unique. Within each ID you will find journal responses, and each has a numerical index assigned to it. This is how the game keeps track of progression through a quest, and through the game. When milestone is reached in a quest, a journal index is called (either through dialog or via a script). Journal responses must be listed in order according to index, but it does not matter if the order start at the top of the list, or bottom. It also does not matter what the index number you start at, nor stop at, as long as the number are progressive (i.e. the next index is higher in value than the last). Thus you could start an index at 1 or 10. You could progress by ones (1, 2, 3, 4, etc), or by 5 (5, 10, 15, 20, etc), or whatever. As long as the next number in the series is higher than the last. I like to work with numbers that progress by 5, that way if I need at a later date and add another index, I can insert it in between where I need to, without having to shift numbers around. But that's just me.

Keep in mind: it is important to remember that journal IDs must be named correctly. The format that Bethesda uses is good (A1_journal_name). If you just use simple words for a journal ID you run the risk of the game thinking your journal entry is actually a topic. Bad juju.

Check your spelling. Pain in the spotted owl, isn't it? But a mod that is rife with misspelled words and bad grammar will make you look bad. But there is an easy way to do this. Use the Export function under the File Menu. Export New Dialog only, then find the .txt file in your Morrowind folder, then open with MS Word or Open Office, and run the Check Spelling option. Once done, import it back into your mod using the Import function under the File Menu. Does it get any easier? I don't think so.

Note: *Do not change the format in any way when you do this! Or else the dialog will not import correctly back into your mod (i.e. it will screw things up. Unless you enjoy rewriting all that dialog).*

Dialog use and Review. So now hopefully, you understand the ins and outs of how to use the dialog window. But what does it all mean? How do I structure greetings, topics? Help! Ok, calm down. Let me try to give you some fundamentals on how to do simple stuff. I won't go into anything complex, nor extrapolate on different situations, but I'll get you started. Now pay attention, this is the closest thing you'll get to a tutorial in this manual.

A standard run and grab quest. First, choose the NPC to give the quest. We'll create one called Tala. Now we create a new object called `special_vase`. Ok, done. Now open the dialog window. We'll make the journal entries first. Click the Journal tab, and within the journal ID list, right-click, and choose New. Let's name the journal ID 'TA1_vase'. POOF! It's done. Find and click on TA1_vase in the list. Now, let's add the journal indexes. Right-click over the empty response field, and choose New. Let's type in the first journal entry where the quest is given, "A dark elf named Tala asked me to go and fetch a vase from the Balmora Mages Guild, it's sitting next to the bunks". Now let's give it an index of 5. Done. Now r-click again, choose New, and type, "I have found the vase right next to the bunks as Tala told me". Give it an index number of 10. This will let the PC know they have found the right vase. Create a last new journal and type, "Tala thanked me for the vase, and even gave me a few coins", add an index of 15. That's it. The journal entries are done. Order of the list can be up or down, but doesn't matter really.

Now we will do the greeting for Tala, and at the same time add a topic for the PC to click on. Since we want Tala to react normally to the PC having a disease, being naked, and that sort of thing, we'll click on the greetings tab and skip down to Greetings 5. Pick anywhere within the middle of the list, r-click and choose New. Type in, "Greetings. I have a task that I need help with. It concerns a special vase I left behind during a journey." Now click on the ID drop menu, find and select Tala from the list. This is so only she will give this greeting, no one else. Now, r-click over this same response, choose New again. Type in, "Do you have the special vase yet?". Assign Tala, then click on a Function drop menu, select `Journal`, in the second drop menu find and select TA1_vase, then `>=`, and put a 5 in the last box. This means the PC will only get this response once they've been given the initial journal index of 5. Now, r-click again over this response, choose New yet again, and type, "Thank you for your help in recovering my special vase." Set the ID to Tala, and add the `Journal` function with the drop menus, but instead set the index in the last box to 15. This greeting now will only be given after the PC has completed the quest. Whew!

Now we create the topic and add responses to it. Click on the Topics tab and r-click over the topics list. Choose New. The topic will be `special_vase`. Now find and click on it in the list. R-click in the response list, choose new, and type, "I need someone to go to the Balmora Mages Guild. Downstairs next to the bunks I left a special vase there when I last visited. If you can, please return that vase to me." Open the ID drop menu and assign Tala. Now go to the Results box and type: `Journal`, "T1_vase", 5. This will give the PC a journal entry when they click on the topic and get this response to start the quest. Now, create a new entry *above* this response. Type, "Did you not find the vase? I really need that vase, please get it for me." Assign Tala, then we need to add the function `Journal`, TA1_vase, `>=`, 5.

Below that click the drop menu, choose `Item`, then click the second drop menu and find the `special_vase`, set the third menu to `<=`, and finally a 0 in the last box. This will check how many `special_vase` items are in the PC's inventory to see if it's equal to or less than 0. Now create another new response above this one. Type, "Thank you for returning my special vase. I only have a few coins, but you may have them." Assign Tala, add a journal function for TA1_vase, `>=`, 10. Below that line add `Item`, `special_vase`, `>=`, 1. Now, add the following lines to the Results box:

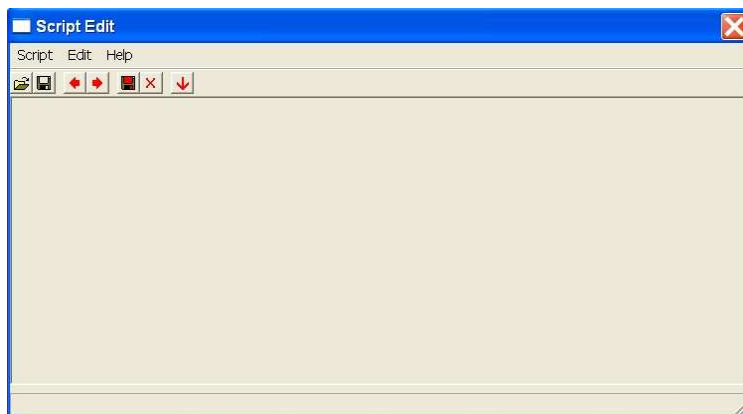
Journal, "TA1_vase", 15
 RemoveItem, "special_vase", 1
 AddItem, "gold_001", 20

This gives the last journal entry completing the quest, removes the vase from the PC's inventory, and gives the PC 20 drakes for their hard earned work. Now, we need to add one more line. Create another line above the last and type, "You were very kind to retrieve my vase. It would have been hard for me to travel and get it myself." For this we only need one condition. Add `Journal, special_vase, >=, 15`. This response is to prevent any of the lower responses being given more than once, or after the quest is complete. That's it for the dialog! You're done! Now, all this assumes you created an object with the ID of `special_vase`, added a script to it so the PC get's a journal entry when they put it in their inventory, and put the vase next to the bunks at the Balmora Mages Guild, and put Tala in the game somewhere also.

Bear in mind this is a very, very simple basic example of how you can do a quest with dialog. Nothing special or unique here. If you are still having trouble working with and understanding dialog, I recommend grabbing a tutorial about it, and/or post for help in a forum of your choosing.

The Scripting Window

Scripting is a whole new world of stress for the uninitiated newbie. It seems daunting at first trying to figure out what all the functions are for, unscrambling and understanding commands and how they go together to make something happen, and the joy of watching your first script produce unexpected, unintentional, and indecipherable results (if any). I will not cover all the console



functions, I have previously listed those that are specific to the console. Most functions can be used in the console, and some can be used in Dialog (which I will cover there). I will cover scripting in three parts; first the 'commands' used in a line to produce a result, second the syntax, or how to put them together with a function to produce a result, and third, functions and what they do (and don't do!). I will commonly refer to commands and functions within a script as 'code' in this section. There are quite a few conventions to keep in mind when scripting:

***Note:** While I am not crediting anyone directly within this work, I cannot fail to mention GhanBuriGhan's **Scripting for Dummies** here. I learned how to script reading his work and kept it open as I scripted, constantly referring to it to clear up problems or misunderstandings. Much of what I put into this section is from my experience scripting, but his work is the base. Even as I type I am referring to his work to make sure I don't type something incorrect or to jog my memory. I would be remiss if I didn't entirely credit him for this section being possible.*

Another great resource is the UESP website by Dave Humphries, he has a comprehensive definitive listing of everything you can use to script with (refer to [Appendix C](#)).

Keep in mind: While I have tried to include as much information as possible this section is intended to give even a newbie a good base to learn scripting, and as a reference for anyone else. But this does not cover everrrrythiinnnggggg. To do that would take much longer, add several hundred pages to this work, and cause my untimely death (my wife is very understanding, but even she has her limits). I suggest you get and keep handy a copy of GBG's work, and mark the UESP site for offline access, for when you need to know something I don't cover.

Making Scripts

- I refer to any script that when written up in the CS as 'big' if you can't see all the code at one time. If you have to scroll up or down to see any part of the code, I consider it big. Anything else is 'small' and easier to work with.
- The Scripting language is not case sensitive (Capital and lower case). However, if you choose to follow the convention used by Bethesda (which is to capitalize each word within a function. Ex: HitAttemptOnMe), or choose to make everything lowercase, stick to it. I suggest using the convention Bethesda uses, it makes reading a script easier, and picking out the functions and commands you are looking for. **BUT** whatever you choose to do, **stick to it**. It makes for cleaner scripts that are easier to read and understand. This is admittedly minor, but will save you some grief.
- **Spaces are a good thing.** When you compile a script a space is ignored, so often you can leave a space out with certain commands BUT (and this is a big but, kinda like mine), you may experience problems with the script if you do so. Thusly, if I mention it is a good idea to use a space somewhere, DO IT. In general it is a good idea to use spaces between all commands and functions. This is a big grief saver.
- **Use quotes** when using an object's ID, if the ID is more than one word and has spaces in it, you **must** use quotation marks around the ID (i.e. "big ass sword" vs. big_ass_sword). Anytime you use a filename in a script you **must** use quotes (i.e. "anymovie.bik").
- **Use a comma** after a function and between parameters (numerical values). Scripts will usually compile and work if you forget to use a comma, but you might have errors if you don't. Rare, but better safe than sorry. With some functions you **must** use a comma afterward, usually those that involve filename use.
- **Set and use 'do once' conditions.** Because a script processes every frame, anything you have a script do, it will do many times each second. You probably don't want a script that acts as a trap on a chest to zap a player 20 times per second with damage, every second, to infinity. Declare a variable called 'doonce', check if the variable is equal to 1 or not, then after a function processes you only want to happen once, add a line that sets the doonce variable to 1. Next frame the script should check the doonce variable before the function, and since set to 1, skip the function. It could look something like this:

```
if ( doonce == 1 )
    Return
else
    [some function]
    set doonce to 1
endif
```

This is just an example. Learn to use a do once statement in your scripts.

- **Write out your script** on paper first, at least a draft. I will put to paper a rough draft, maybe leave out a lot of the code even, just so I have a good working idea of how easy/hard the script will be, and identify any problems that may arise. Some do this, some don't. I usually don't write much code, just the concept of each section of a script. This can help you when working with big scripts.
- **Divide your script into sections**, and then fill it in with the necessary code. When I write out the concept of a new script I divide it up into sections. Each section of the script is accomplishing something for the script as a whole. One section may only be one or two lines, maybe compute some numbers for a variable, or maybe nothing more than checks for data. Whatever. This will help you make sure all parts of a script are doing what they are supposed to do when writing it out, and when debugging it. Trust me, for a big script, it's a grief saver.
- **Use the semicolon (;)** to comment on what you are doing in each section of a script. Especially if you are just learning how to script. This will keep you focused on what you are trying to do with the code of that section. In a big script it can be easy to loose track or forget what you were trying to do exactly with a given section.
- To save a script you must click the 'compile' button. If there are no errors in your code it will save your script. If there are errors, it will not compile and will alert you to the fact something is wrong (though not necessarily what). You will get an error statement when you try to compile for the first error found. So you may have to click compile, fix an error, hit compile again, fix another error, hit compile again,... etc. Just because you don't have an error when you compile a script doesn't mean it will work, the engine just checks to make sure there are no syntax errors. Your script might still be crud (and this is why we test our plug-ins people).

***Note:** Compiling a script only saves the form of the script, this does not save it in the CS. You **must** still click save (from the menu or toolbar) for the script to be saved as part of your plug-in.*

How Scripts Work

- **Scripts run once every single frame** (you remember what frames are right?). This means a script may execute or process 10, 20, 40, or more times every second. Also, frame rates vary from computer to computer, and second by second on any single computer. Keep this in mind.

***Note:** The game is notorious about running scripts. It is quite common that scripts get skipped some frames. The more the game has to process each frame, the more likely a script will get dropped that frame. This means out of 10 frames, a script might get dropped once or twice. Not a real big deal, but something to just be aware of.*

- There are two types of scripts: Local and Global. Local scripts are any script that's attached to an object. Local scripts will automatically start the moment the PC enters the cell (in an exterior cell, the moment that cell is loaded, even if the PC isn't 'in' it yet).

Global scripts must somehow be started, are not attached to anything, but process regardless of what cell they are in (according to their code). You must use another script or dialog to start a global script or to stop it. Some global scripts start at the beginning of the game to set global variables and set how the game is to run (called startup scripts). Don't mess with these, trust me, just don't.

- Local scripts, since they are attached to an object, usually affect only that object with their code unless you specify differently (within that script). This means you don't have to state the object it's attached to as the object to be affected by a function, it's automatic.
- Global scripts, since they are not attached to any object, must have specified within their code, what they are affecting with any function stated.
- Scripts use variables to *hold* or *remember* data (and you thought algebra would never come in handy back in school). The variable is a single word (or several together without spaces OR connected with an underscore). They *hold* data, or numbers. You must list (or *state*) variables you will use in a script at the start of a script before any code (but after the `Begin` command). This is referred to as '*declaring*' a variable in a script. Most of the time, a variable will have a value of 0 (zero) until you assign data in the script.
- Variables come in three flavors; Long, Short, and Float. You must state which type a variable is supposed to be when you declare it. I have yet to see a Long variable declared in a script, but you can.
- **Long variables** can be assigned a value anywhere from -2,147,483,648 to 2,147,483,647.
- **Short variables** can be assigned a value anywhere from -32,768 to 32,767.
- **Float variables** can be assigned a value anywhere from 3.4E +/- 38. What does this gobbledygook mean? It means the variable can accept a numerical value that has a decimal in it, with up to seven digits after the decimal.
- Once a value is assigned to a variable it will always have that value until changed, no matter if a local or global variable, what cell you are in, exiting the game, or turning your computer off. Exception: if you use the [StopScript](#) function any variables are reset.
- In a local script, you can pretty much make up any name for a variable you want, even include numbers, as long as the variable starts with a letter. The variable will only work with that script. Try to make it memorable, and easy to identify what data the variable holds (ex: if you use a variable to hold the current Strength of the PC, name the variable 'currentstrength' or 'currstr'). Try not to accidentally use function names, and don't use punctuation. Both will really screw things up. Be creative. Don't name a local variable the same as a global variable. The global variable will override the local, meaning the game will use the global value instead of the local.
- The game already has some global variables declared, you cannot change them (without serious game breaking repercussions). You can use them in a script. Any variable you declare in a global script becomes a global variable.

The Commands. These are the words and symbols used in a line of the script to state how the script processes and what is affected by it.

Begin. This is the command that states where a script starts. You put the name of your script on the same line after this command. You must put a space between this command and the script name. The script name cannot have any spaces. Thus either of the following is a valid way to write the first line in a script:

Begin CatNightEye OR Begin Cat_Night_Eye

This is a no-no:

Begin Cat Night Eye OR BeginCatNightEye

Note: *You cannot put anything on a line above this command, except text after a semicolon. Don't do this though, just put any descriptive text below this command. Keep things simple.*

End. This goes on it's own line at the very end of a script. This tells the game no more code for this script, it's done for this frame. When this command is reached the game goes back to the top of the script and waits for the next frame to execute.

Semicolon. Use a semicolon (;) to add text. Any text you place after the semicolon on the same line will be ignored when the script is compiled and processes ingame. This doesn't mean it's deleted, it stays where you put it, its just ignored. You cannot use the 'enter' key for a carriage return! Only the text placed after the semicolon and on the same line is ignored. If the text wraps around to the next line, that's ok. Thus, this is ok:

```
GetStrength->player      ;get the player's current strength for testing against item
weight
```

But not this:

```
GetStrength->player      ;get the player's current [carriage return]
[tab][tab][tab] strength for testing against [carriage return]
[tab][tab][tab] item weight
```

If. You use this command to check if some aspect of the game meets a condition you have stated. Thus:

```
If ( currentstrength == 30 )      ;checks for a Strength exactly equal to 30
```

The `if` command checks to see if the function statement following it is true or false. If true, it will process any functions listed below it until another command statement is reached. If false, it will skip any functions until another command statement is found below it.

Elseif. Use this below an `if` statement to check for more than one condition. Thus:

```
If ( currentstrength == 30 )      ;checks for Strength equal to 30
[some function]
Elseif ( currentstrength == 40 )      ;checks for Strength equal to 40
[a different function]
```

Think of the `elseif` command as similar to a Boolean 'and' or 'or'. It otherwise works just as does the `if` command.

Note: *if you use multiple elseif statements within an if/endif set, the game does not check every elseif statement, it will accept the first elseif statement that returns true, and process nested functions within it.*

Else. Use this after any `if` or `elseif` commands to check a final or 'catch all' threshold or condition. Thus:

```
If ( currentstrength == 30 )           ;checks for Strength equal to 30
    [some function]
Elseif ( currentstrength == 40 )       ;checks for Strength equal to 40
    [a different function]
Else                                   ;if Strength is not equal to 30 nor 40
    [entirely different function]
```

Think of the `else` command as 'if none of the above'. If no `if` or `elseif` conditions above it return true, this command will process any functions listed below it until another command is reached.

Endif. You must use this command to finish any (set of) `if` statement(s). The script will not compile without all `if/elseif` statements ending with an `Endif` command. Thus:

```
If ( currentstrength == 30 )           ;checks for Strength equal to 30
    [some function]
Elseif ( currentstrength == 40 )       ;checks for Strength equal to 40
    [a different function]
Else                                   ;if Strength is not equal to 30 nor 40
    [entirely different function]
Endif
```

You can also embed `if` statements within other `if` statements. The indented `if` statement is said to be embedded or nested within the first `if` statement (There is reported to be a maximum of ten embedded `if` statements within a singular `if` statement. Thus:

```
If ( currentstrength == 30 )           ;checks for Strength equal to 30
    If ( currentagility == 30 )       ;checks for Strength and Agility both
        [function]                   ;equal to 30
    Endif
    [some function]                   ;if Strength is 30, but agility is not 30
Elseif ( currentstrength == 40 )       ;checks for Strength equal to 40
    [a different function]
Else                                   ;if Strength is not equal to 30 nor 40
    [entirely different function]
Endif
```

This is a simple `if` set of statements. Note the structure. The `if` command comes first, then any `elseif` commands (optional, you may not need to use any. You can use more than one if you need to, quite a few in fact), then the `else` command (optional, you don't have to use this command, but useful when you only need to check for a few conditions out of many). Then the `endif` command to end the set. Notice how I indented the [function statement]. The function is embedded in the `if` statement and only gets processed when the `if` statement returns (or is) true. It is not necessary to indent, but is oh so useful when scanning through lines of code trying to understand what you have done (and/or done wrong).

set/to. Use this pair to assign a value to a variable:

```
set currentstrength to ( GetStrength->player ) Or Set currentstrength to 1
```

While. It works similar to the `if` command. However, unlike the `if` command that just checks to see if a statement returns true or not once per frame, this command will force a script to stop in place (and the game), and process an embedded function **until** the `while` statement returns true. Be advised, this can seriously impact gameplay, causing lags in FPS, or even causing the game to hang while the script processes until the `while` statement returns true. You've been warned. Why use it then? Due to certain limitations of the scripting language there may be times when it is easier and beneficial to use the `while` command. Still, use this command with caution.

Endwhile. This command is used to finish the `while` statement. Works just like `endif`.

Syntax.

-> This arrow is used to designate the calling object, and the condition or function to apply to it. The calling object is usually the `object_ID` on the left side of the 'arrow'. The condition to check or function to apply is usually stated to the right of the 'arrow'. Thus:

```
object_ID->GetStrength OR player->SetStrength, 50
```

Note: The object 'ID' of the PC is always 'player'.

The function can also be placed on the left and the object's ID on the right. Thus:

```
SetStrength->player, 50
```

But not:

```
GetStrength->NPC_ID
```

Why? In the first you are applying the function to the PC, the numerical value can be placed after the object as so. But in the second you are calling for what that numerical value is, but cannot assign it to a variable with the given statement syntax. In the above statement you're telling the script to 'get' the Strength of the NPC but cannot tell the script what to do with it.

When you state an object's ID in a script, the game will look for the first such `object_ID` listed in its database. Which means you might not get the object's reference you want the script to work with. This is why it's a good idea to give an object a unique name that you want to attach a script to.

Keep in mind: With a local script if you don't use the 'arrow' but just state a function, it will automatically be applied to the object the script is attached to. In a global script you *must* state the object ID any function will be applied to (i.e. you must use the 'arrow' to designate the objects involved).

Note: Some functions have specific syntax you must use slightly different from that above.

Condition Check signs

- `==` This is used to set a condition of 'is equal to'.
- `!=` This is used to set a condition of 'is not equal to'.
- `<=` This is used to set a condition of 'is less than or equal to'.
- `<` This is used to set a condition of 'less than'.
- `>=` This is used to set a condition of 'is greater than or equal to'.
- `>` This is used to set a condition of 'is greater than'.

These are used as thus:

```
If ( player->GetStrength >= 30 )      ;if the PC's Strength is equal to or greater than 30
    [function]                        ;do this
endif
```

Note: Almost always when you make an `if` statement you will need to use parentheses to encapsulate the condition check part of the statement (i.e. the part with the '`->`' arrow). Whenever you use parentheses you should make sure to put a space on both sides of each parentheses. Don't argue, DO IT! Also make sure to put a space before and after the condition setting signs (i.e. `==`, `!=`, `<=`, `<`, `>=`, `>`). If you have to ask why, see the above bullet about using spaces in code.

Mathematics

- `+` This is used for addition.
- `-` This is used for subtraction.
- `*` This is used for multiplication.
- `/` This is used for division.

Math is done just like you learned in school. Thus:

```
Set variable_x to ( variable_y + variable_z ) OR
Set currentstrength to (( variable_a + variable_b ) * (( variable_c - 10 ) / 5 ))
```

Note: the rule about spaces applies to math symbols too (a space on each side).

Note: I have seen several statements saying that there is a maximum number of variables in math calculations that can be done on one line of code. Supposedly 20, I have not tried to exceed this to see if true, but recommend you split big calculations between several lines to avoid the 'rule of 20'.

Let's review Commands. Each script must have a `Begin`, then code, then `End`. You use `if/endif` statements to test conditions. You can use `set/to` to assign numerical values to variables, or to assign one variables value to another variable. You use variables to hold these values within a script and/or to manipulate them. You must declare variables. Most variables start at a value of 0 before assigned a value. Use spaces. Here is a sample script (below) so you can see a complete format. The script begins, declares a variable, checks if this script has ever processed before, tests if Strength is under 40, then tests if Agility is under 40, and if so, sets Strength to 40. If either Strength or Agility is over 40 then no changes are made. After the `if` statements, the variable is changed to a 1. The script then ends and starts processing again. Now, on the second run through the script the game gets

to line 5, checks the variable's value which was set to 1 last frame. Since it is 1, greater than 0, the game skips the entire set of if statements, ends, and starts over again the next frame, ad infinitum.

Begin ASampleScript	;script starts processing here
short variable_one	;declared a variable
if (variable_one < 0)	;has this script processed before?
if (player->GetStrength < 40)	;is the PC's Strength less than 40?
if (player->GetAgility < 40)	;is the PC's Agility less than 40?
Player->SetStrength, 40	;if not, change Strength to 40
Endif	
Endif	
Endif	
Set variable_one to 1	;change variable_one to 1
End	;script ends here

Note: this script is just to show format.

The Functions. You're still not ready to start scripting yet, you need to know about functions and what they do. There are many, many functions. Most are useful, a few have limited use, and several just plain don't work. I will list each function, show the syntax to use it, and a brief description. Let's get to the functions.

Activate

Activate

This will cause an object to do whatever it does as if the spacebar had been pushed. You must use this function for an object after using **OnActivate** if you wish the object to continue with it's normal function.

NPC	Will initiate dialogue
Container	Will open
Door	Should open, but doesn't seem to work
Weapon, armor, etc.	Will be picked up
Book/Scroll	Will open up to be read

Note: You can place `player` after the `Activate` function in a script. This can be useful to have the PC access an object in another cell by placing the script on an object in inventory, and having the object to be accessed as the calling object. The accessed object must have the *references persist* box checked. This would look like this:

```
Object_ID->activate, player
```

This will only work though if the object called is loaded in game memory (i.e. has been activated) during the current game session.

AddItem

AddItem object_ID, [number]

This will add the stated number of objects to an inventory.

Note: *if you add an item to inventory, any script attached to that item will not process until the item 'enters' the game world. Usually by dropping the item on the ground (and then picking it back up).*

AddSoulGem

AddSoulGem creature_ID, soulgem_ID

This will add one stated soulgem to the inventory with the stated creature's soul trapped inside.

AddSpell

AddSpell spell_ID

If used on an PC or actor this will add the stated spell to the spell inventory. Spells, Abilities, and Powers are added to the appropriate place list. Diseases and Curses will affect the PC or actor.

Note: *If a curse or a disease is added to a creature the game will add it to all references of that creature from that point onward. Suggest you use a [RemoveSpell](#) function with an [OnDeath](#) function to remove the curse/disease upon the creature's death to prevent this (unless you want it).*

AddToLevCreature (Tribunal)

AddToLevCreature leveledcreaturelist_ID, creature_ID, [level]

This will add the stated creature ID to the stated leveled creature list ID, to appear at the stated minimum level. If a leveled list has generated a creature already, the added creature can not appear until the next time the list generates a creature, even though it is added to the list.

AddToLevItem (Tribunal)

AddToLevItem leveleditemlist_ID, item_ID, [level]

This works exactly the same as AddToLevCreature.

AddTopic

AddTopic topic_ID

This will add a topic to the known topics that the PC knows about and can discuss in dialog. Until you add a topic somehow to the PC's known discussable topics, the PC will not see them show up in conversation with an actor, even if the actor has the topic as part of their conversation. Use this function to introduce a topic for discussion.

AIActivate (updated with Tribunal)

AIActivate object_ID, [reset]

This will cause an actor to activate an object (though you could just use the [Activate](#) function instead for some actions). Refer to the list under Activate for action taken. There does seem to be a problem with doors that teleport (i.e. move the PC to a different cell). This function will only work properly with such doors when the actor/PC is moved to a door in the same cell (interior or exterior). Also, this function was updated with Tribunal. In the original Morrowind game this function would only force an actor to drink a potion, it would not otherwise work. The reset does not appear to work and seems to just cause problems (so don't use the reset).

AIEscort**AIEscortCell**

```
AIEscort actor_ID, [duration], x, y, z, [reset]
AIEscortCell actor_ID, cell_ID, [duration], x, y, z, [reset]
```

This function makes an actor move to the x, y, z grid specified, but they will only actually move when the specified actor/PC follows them. If the escorted actor/PC stops following, the escorting actor will stop in place and wait for them to 'catch up'. If you set duration to zero, there is no time limit and the function ends when the destination is reached. If you set duration to zero and the x, y, z grid to 0, 0, 0, the actor will escort along a given path until a condition you set is met or completed. If the escorted actor/PC is attacked at any time the escortee will defend them. The reset flag function is unknown.

EscortCell works exactly the same except the escort will stop escorting in the specified cell and grid. For working with interior cells.

***Note:** Make sure you only issue this function once to an actor or you can crash the game.*

AIFollow**AIFollowCell**

```
AIFollow actor_ID, [duration], x, y, z, [reset]
AIFollowCell actor_ID, cell_ID, [duration], x, y, z, [reset]
```

Works the same as AIEscort except now the actor follows the NPC/PC.

***Note:** Make sure you only issue this function once to an actor or you can crash the game.*

AITravel

```
AITravel x, y, z [reset]
```

This causes an actor to move to the specified grid. If a Path Grid is present they may use it, but it is not necessary. If you try to move the actor to another cell you may experience problems. If the PC sleeps, uses fast travel, or teleports, the actor will warp instantly to the specified grid (not a big deal in and of itself), but [GetAIPackageDone](#) will not detect completion.

***Note:** Make sure you only issue this function once to an actor or you can crash the game.*

AIWander

```
AIWander [range], [duration], [time], [idle1], [idle2], [idle3], [idle4], [idle5],
[idle6], [idle7], [idle8], [idle9], [reset]
```


This will make an actor move about the cell in a random pattern. If a path grid is present they may use it. An actor will not move farther from their starting point than the specified range (in game units). Duration is amount of time spent wandering in hours from start time. Time is the hour of the day they begin wandering. If all three of these are set to zero, the actor will remain in place. Each idle is a percentage change (expressed as a value of 1-100) of that action taking place. Each second a roll is made vs. each idle, the one with the highest passing roll takes place. If none pass the actor walks. If the actor is set to not move, they merely stand and stare. The actions associated with each idle are listed below. The reset flag does not seem to work.

- Idle1: Stand still
- Idle2: Shift weight from one leg to another
- Idle3: Looking behind
- Idle4: Scratching or shaking head
- Idle5: Shifting clothing or armor on shoulder (females put hand on hip instead)
- Idle6: Yawning and stretching
- Idle7: Looking at fingers and looking around furtively
- Idle8: Putting hand on chest (as if having heartburn)
- Idle9: Reaching for weapon, then touching head (Khajiit females scratch and shake head)

***Note:** There are two known problems with this function when used in a script; the Idle2 data doesn't seem to get compiled, and the reset will cause problems if used.*

BecomeWerewolf (Bloodmoon)

Player->BecomeWerewolf

This will turn the actor/PC into a werewolf. For NPCs they will take on the werewolf mesh and appearance, but will retain the same stats (including AIPackages, Fight, Flee, Hello settings, etc). This means they won't attack the PC or another creature unless they would have anyway (or you make them). They lose their equipment when in this form and cannot participate in dialog with the PC. You can use this function on some creatures too, strangely enough. They still look the same, but act as a werewolf. Though they will retain their AIPackages, Fight, Flee settings, etc.

***Note:** If you use this function on the PC prior to them acquiring the werewolf disease normally in the game, or after they have cured it, you could break the game's quests. This function will change the global variables associated with lycanthropy, and some quests check those variables to work properly. Use this with caution.*

Cast

Cast spell_ID, target_ID

This will cause an object to cast the specified spell at the specified target. This only works with spells, not abilities, powers, etc.

CellChanged

```
if ( CellChanged == 1 )
    [something happens]
endif
```

When the PC *exits* the current cell this function will return a value of 1 for one frame. If used in a local script this will return a value of 1 for one frame when the PC *enters* an interior cell containing the object this script is attached to. I've had trouble with this function being processed at times in cells (Not sure why, but may be related to tough scripts or objects being rendered by the game).

Note: *There is a reported problem with this function not returning a value of 1 if the PC teleports out of a cell.*

CellUpdate (BROKEN)

This function is supposed to update an object's position when they move from one cell to another as the game stores data about objects based on the cell they start in. This is supposed to update that data. If you use this function it will generate errors and/or crash the game.

CenterOnCell

```
CenterOnCell, cell_ID
(COC can be used in the console)
```

This will place an object in the center of a cell. Remember to use quotes ("") if a cell name has spaces in it.

CenterOnExterior

```
CenterOnExterior, cell_ID
(COE can be used in the console)
```

Same as CenterOnCell, but for exterior cells.

ChangeWeather

```
ChangeWeather region_ID, [weather value]
```

This will change the weather to that specified by the stated value after the region ID. The weather will update and change normally again after 20 hours (this is set in the morrowind.ini file). The value to enter for the new weather type is:

- 0 = Clear
- 1 = Cloudy
- 2 = Foggy
- 3 = Overcast
- 4 = Rain
- 5 = Thunder
- 6 = Ash
- 7 = Blight
- (8 = Snow?)

(9 = Blizzard?)

ClearForceJump (Tribunal)
ClearForceJump

This will stop an actor from jumping after being forced to jump with the [ForceJump](#) function.

ClearForceMoveJump (Tribunal)
ClearForceMoveJump

This will stop an actor from jumping while running after being forced to do so with the [ForceMoveJump](#) function.

ClearForceRun (Tribunal)
ClearForceRun

This will stop an actor from running after being forced to run with the [ForceRun](#) function.

ClearForceSneak
ClearForceSneak

This will stop an actor from sneaking after being forced to sneak with the [ForceSneak](#) function.

Disable
Disable

This will stop the game from rendering an object. It still exists, and if a script is attached to it, the script will still process as normal, but the object can neither be seen nor touched (it isn't there).

***Note:** You cannot disable an object from a script attached to it, this will crash the game. Use a different script to do this. Disabling lights also seems to sometimes cause problems (the light object is gone, but 'light' is still emitted and rendered).*

DisableLevitation (Tribunal)
DisableLevitation

This will disable all levitation.

DisablePlayerControls
DisablePlayerControls

This will disable movement, menus, and view switching ability of the PC. All the PC will be able to do is swivel their viewpoint in 1st person mode. This is how the PC starts the game.

DisablePlayerFighting

DisablePlayerFighting

This will prevent the PC from equipping weapons. If a weapon is equipped in hand already they can still use it. Quick-keys will still equip a weapon.

DisablePlayerJumping

DisablePlayerJumping

This will prevent the player jumping.

DisablePlayerLooking

DisablePlayerLooking

This prevents the player from swiveling their point of view as if paralyzed.

DisablePlayerMagic

DisablePlayerMagic

This will prevent the player from casting any spells. If a spell is already readied, it can still be cast. Quick-keys can still be used to cast spells.

DisablePlayerViewSwitch

DisablePlayerViewSwitch

This will prevent the player from switching between 1st and 3rd person view.

DisableTeleporting

DisableTeleporting

This will prevent the PC from teleporting via the Recall spell. This does not prevent use of doors or fast travel.

DisableVanityMode

DisableVanityMode

This will prevent the PC from switching to 3rd person view.

DontSaveObject

DontSaveObject

This will prevent the game from saving any changes made to the object (ingame) in a save game file.

Drop

Drop object_ID, [number]

This will drop the stated number of the specified object from an actor/PC's inventory to their feet. It will not function if you stipulate more to be dropped than is present in the inventory.

***Note:** when items are dropped by an NPC they will appear at the PC's feet instead of the NPC.*

DropSoulGem
DropSoulGem creature_ID

This will cause a single soulgem filled with the specified creature to be dropped from an object.

Enable
Enable

This enables an object that is not being rendered due to the [Disable](#) function used on it.

EnableBirthMenu
EnableBirthMenu

This will bring up the birthsign choice menu for the PC to pick from.

EnableClassMenu
EnableClassMenu

This will bring up the class choice menu for the PC to pick from.

EnableInventoryMenu
EnableInventoryMenu

This will allow the PC to access their item inventory menu if it has been disabled.

EnableLevelupMenu
EnableLevelupMenu

This brings the level up menu to appear when the PC rests and meets the requirements to go up a level. There is no corresponding disable function.

EnableLevitation (Tribunal)
EnableLevitation

This will allow the PC to use levitation magic if it has been disabled.

EnableMagicMenu
EnableMagicMenu

This will allow the PC access their spell inventory menu if it has been disabled.

EnableMapMenu
EnableMapMenu

This will allow the PC's large map to display with the menus. Not sure what function would disable this, suspect it to be [DisablePlayerControls](#) (only).

EnableNameMenu
EnableNameMenu

This will bring up the name choice menu for the player to choose their PC's name.

EnablePlayerControls
EnablePlayerControls

This will allow the PC to access all menus, view modes, and movement abilities.

EnablePlayerFighting
EnablePlayerFighting

This will allow the PC to equip weapons if the ability was disabled.

EnablePlayerJumping
EnablePlayerJumping

This will allow the PC to jump if the ability was disabled.

EnablePlayerLooking
EnablePlayerLooking

This will enable the PC to swivel their viewpoint around if it was disabled.

EnablePlayerMagic
EnablePlayerMagic

This will enable the PC to cast spell if the ability was disabled.

EnablePlayerViewSwitch
EnablePlayerViewSwitch

This will allow the PC to switch between 1st and 3rd person views if it was disabled.

EnableRaceMenu
EnableRaceMenu

This will bring up the Race choice menu allowing the PC to pick a race.

EnableRestMenu
EnableRestMenu

This brings up the rest menu so the PC can rest. There is no corresponding disable function.

EnableStatReviewMenu
EnableStatReviewMenu

This will bring up a stat menu for the PC to review their choices of birthsign, class, name, and race.

EnableStatsMenu
EnableStatsMenu

This will allow the PC to access their stat menu if it has been disabled.

EnableTeleporting
EnableTeleporting

This will enable teleporting via the Recall spell if it has been disabled.

EnableVanityMode
EnableVanityMode

This will allow the PC to switch to 3rd person view if it has been disabled.

Equip
Equip object_ID

This will make the actor/PC equip the specified object. If the object is not in the actor's inventory, the game will add the object (as per [AddItem](#)).

***Note:** This function apparently was broken somewhat in the original Morrowind game but was mostly fixed with Tribunal.*

ExplodeSpell (Tribunal)
ExplodeSpell spell_ID

This spell will make an object cast a spell at itself. Use an area effect touch ranged spell and the reference can 'explode'. Prior to Tribunal you had to have an object [Cast](#) the spell with the target as itself.

Face
Face [angle], [time]
Face x, y

To use this function either state the function, angle in degrees to turn and face, and amount of time it takes to make the facing movement OR state the function and enter the x, y coordinates to be faced. Apparently when used, the actor will stop, turn and face as specified, then continue on doing whatever they were doing.

FadeIn
FadeIn [time]

This will cause the screen to fade from a black screen to normal in the stated amount of time of 0 to 10 seconds.

FadeOut
FadeOut [time]

This will cause the screen to fade to a black screen from normal in the stated amount of time of 0 to 10 seconds.

FadeTo

FadeTo [% opaque], [time]

This will cause the screen to fade to a stated percentage of 'blackness' in the stated amount of time. Zero is a normal screen, 100 is a totally black screen.

Fall

Fall

I have not tested this function, but assume it would cause an object to fall to the next object below it.

FixMe

FixMe
(Can be used in the console)

This will move your PC 128 units (or 71.68 inches). Good for getting unstuck. Use this in scripts when moving an NPC/actor over long distances (through cells). May help with clipping issues that can arise.

ForceGreeting

ForceGreeting

Use this and an NPC will begin dialog with the PC. This will happen no matter where the NPC or the PC are in the world. This is best used with [GetDistance](#) to ensure dialog doesn't begin till the PC and NPC are near each other (unless you want them to communicate over long distance). This will only work if the PC has entered the same cell as the actor within the last 72 hours.

***Note:** To get around this '72 hour' limit you can use the function [PositionCell](#) once each day. Some report you must use the function to move the NPC into the same cell as the PC, and then move them back to their original position. Others report you can use it with their current/original grid coordinates and it works fine. Have not verified which, if either, is correct.*

ForceJump (Tribunal)

ForceJump

This will cause the actor to begin jumping.

ForceMoveJump (Tribunal)

ForceMoveJump

This will cause the actor to run and begin jumping.

ForceRun (Tribunal)

ForceRun

This will cause the actor to run.

ForceSneak
ForceSneak

This will cause the actor to begin sneaking.

Get-
GetAgility
GetAcrobatics
GetArmorBonus, etc.
[float value returned]

While many functions have `Get` as part of their syntax name, you can use `Get` with any stat, and a few other game aspects too. These are listed below. `Get` will return the current value of any stat (or aspect of the game) when used. Some of the non-stat functions I will describe immediately afterwards instead of alphabetically like other functions. Most of these listed here you can also use `Set-/Mod-` with.

Agility	Acrobatics	ArmorBonus
Alarm	Alchemy	AttackBonus
Disposition	Alteration	Blindness
Endurance	Armorer	CastPenalty
Fatigue	Athletics	Chameleon
Fight	Axe	FactionReaction
Flee	Block	Flying
Health	BluntWeapon	Invisible
Hello	Conjuration	Paralysis
Intelligence	Destruction	PCCrimeLevel
Level	Enchant	PCFacRep
Luck	HandToHand	PCVisionBonus
Magicka	HeavyArmor	ResistBlight
Personality	Illusion	ResistCorprus
Reputation	LightArmor	ResistDisease
Speed	LongBlade	ResistFire
Strength	Marksman	ResistFrost
Willpower	MediumArmor	ResistMagicka
	Mercantile	ResistNormalWeapons
	Mysticism	ResistParalysis
	Restoration	ResistPoison
	Security	ResistShock
	ShortBlade	Scale
	Sneak	Silence
	Spear	SuperJump
	Speechcraft	Swim Speed
	Unarmored	WaterBreathing
		WaterLevel

GetAlarm`GetAlarm [float value returned]`

This will return the current numerical value of the [Alarm](#) rating of an actor. It is not used for the PC.

GetArmorBonus`GetArmorBonus [float value returned]`

I have not tested this function yet, so I am unsure of its value or use. I assume it is a bonus somehow generated and applied or combined with the AR value of armor.

GetAttackBonus`GetAttackBonus [float value returned]`

I have not tested this function yet, so I am unsure of its value or use. I assume it is a bonus somehow generated and applied or combined with the attack value of a weapon.

GetBlindness`GetBlindness [float value returned]`

I haven't tested this, but I assume this will return a value of the percentage of blindness affecting an actor/PC (0-100%).

GetCastPenalty`GetCastPenalty [float value returned]`

I haven't tested this but assume it returns a value of any penalty applied to your chance to successfully cast a spell.

GetChameleon`GetChameleon [float value returned]`

This will return a value of the amount of chameleon (as per effect) is affecting an actor/PC.

GetDefendBonus`GetDefendBonus [float value returned]`

I assume this returns a value stating the bonus applied to your chance to avoid an attack. I have not verified this.

GetDisposition`GetDisposition [float value returned]`

This will return a value of the current [Disposition](#) of the calling actor towards the PC.

GetFactionReaction

`GetFactionReaction faction1_ID, faction2_ID` [short value returned]

This will return a reaction value of stated faction1 against faction2.

GetFight

`GetFight` [float value returned]

This will return the current numerical value of the [Fight](#) rating of an actor. It is not used for the PC.

GetFlee

`GetFlee` [float value returned]

This will return the current numerical value of the [Flee](#) rating of an actor. It is not used for the PC.

GetFlying

`GetFlying` [short value returned]

I assume this returns a value of 1 if actor/PC is levitating, 0 otherwise.

GetHello

`GetHello` [float value returned]

This will return the current numerical value of the [Hello](#) rating of an actor. It is not used for the PC.

GetInvisible

`GetInvisible` [short value returned]

This will return a value of 1 if object is invisible, 0 otherwise.

***Note:** Prior to Tribunal this function was spelled as `GetInvisibile`. Tribunal fixed this.*

GetParalysis

`GetParalysis` [short value returned]

This will return a value of 1 if calling actor is paralyzed, 0 otherwise.

GetPCCrimeLevel

`GetPCCrimeLevel` [float value returned]

This will return a value showing how much gold has been placed on the PC's head for crimes committed.

GetPCFacRep

`GetPCFacRep faction_ID` [short value returned]

This will return the value of the reaction the stated faction has with the PC.

GetPCVisionBonus

`GetPCVisionBonus [float value returned]`

I have not verified it's use, but believe this 'bonus' is the ability to see better with the night eye spell effect.

GetReputation

`GetReputation [float value returned]`

This will return a value equal to the reputation of an actor's starting reputation or the current PC's reputation.

GetScale (Tribunal)

`GetScale [float value returned]`

This will return a value of the current size (as a decimal) of an object.

GetSilence

`GetSilence [float value returned]`

I believe this will return a value that is applied as a penalty to cast spells, but not sure. I think this would result from the Silence spell effect.

GetSwimSpeed

`GetSwimSpeed [float value returned]`

This should return the value of the current speed at which an object swims. I do not know yet if you can set this separately from the Athletics skill (I bet you can), so you can have an actor that can swim like crazy yet has a low athletic skill (and if you increase Athletics, then will the actor then get a likewise increase to their already pumped up swim ability?). This function could have some potentially useful qualities.

GetWaterLevel

`GetWaterLevel [float value returned]`

This will return a value of the current grid coordinate on the z axis the water level is set to.

GetAIPackageDone

`GetAIPackageDone`

This will return a value of 1 for one frame when the current AIPackage completes. Otherwise it returns a value of 0.

GetAngle

`GetAngle [axis] [float value returned]`

This will return the degree (of 360) on the specified grid for the calling object. Zero is north.

GetArmorType (Tribunal)

GetArmorType [armor location] [short value returned]

This is useful for checking what type of armor is being worn by an actor/PC. The armor location value stated determines the body part checked:

- 0 = Helmet
- 1 = Cuirass
- 2 = Left Pauldron
- 3 = Right Pauldron
- 4 = Greaves
- 5 = Boots
- 6 = Left Gauntlet
- 7 = Right Gauntlet
- 8 = Shield
- 9 = Left Bracer
- 10 = Right Bracer

This will return a value for what type of armor is being worn on that body part:

- 1 = Unarmored
- 0 = Light Armor
- 1 = Medium Armor
- 2 = Heavy Armor

GetAttacked

GetAttacked [float value returned]

The will return a value of 1 if the PC has ever attacked the actor, or 0 otherwise.

GetBlightDisease

GetBlightDisease [short value returned]

This will return a value of 1 if the actor has a blight disease, 0 if none.

GetButtonPressed

GetButtonPressed [short value returned]

This function is used in conjunction with the [MessageBox](#) function when offering choices to the PC to pick from. This tests against a button 'choice' pressed and triggers a result based on the button choice made. This will return a value of -1 if no choice has been made yet. It will return a value of 0 for the first button pressed, 1 for the second, 2 for the third, etc.

GetCollidingActor (Tribunal)

GetCollidingActor [short value returned]

This will return a value of 1 if an actor is colliding with the calling actor, 0 otherwise.

GetCollidingPC (Tribunal)

`GetCollidingPC` [short value returned]

This will return a value of 1 if the PC is colliding with the calling actor, 0 otherwise.

GetCommonDisease

`GetCommonDisease` [float value returned]

This will return a value of 1 if an actor or the PC has a common disease, 0 otherwise.

GetCurrentAIPackage

`GetCurrentAIPackage` [short value returned]

This will return a value for the current AI package being executed by an actor. The value returned is:

- 1 = None
- 0 = Wander
- 1 = Travel
- 2 = Escort
- 3 = Follow
- 4 = Activate
- 5 = Pursue

GetCurrentTime

`GetCurrentTime` [float value returned]

I have not tried to use this, but it would stand to reason it either returns the hour of the day (same as `GameHour`), or it works the same as [GetSecondsPassed](#). I would just skip this function and use either [GameHour](#) or `GetSecondsPassed`.

GetCurrentWeather

`GetCurrentWeather` [short value returned]

This will return a value for the current weather (like duh...):

- 0 = Clear
- 1 = Cloudy
- 2 = Foggy
- 3 = Overcast
- 4 = Rain
- 5 = Thunder
- 6 = Ash
- 7 = Blight
- (8 = Snow?)
- (9 = Blizzard?)

GetDeadCount

`GetDeadCount` actor_ID [short value returned]

This will return a value of the number of times stated creature or actor has been killed by the PC.

GetDetected

GetDetected actor_ID [short value returned]

This will return a value of 1 if the calling actor can detect stated actor or PC, 0 otherwise (this would include if the actor or PC is invisible or using chameleon).

***Note:** The TESCS help file states this is a slow function and not to use this very often. Slow? As in it eats up system resources to process? Or just takes a while to return a result? Whichever, use with caution.*

GetDisabled

GetDisabled [short value returned]

This will return a value of 1 if calling object is disabled, 0 otherwise.

GetDistance

GetDistance object_ID [float value returned]

This will return a value in game units of the distance between calling object and stated object. This value will be based upon the original starting grid of an object, so if it has moved at all, use [GetPOS](#) instead. This will also return the first listed instance of the stated object listed in the database, best used with unique objects.

GetEffect

GetEffect spelleffect_ID [short value returned]

This will return a value of 1 if the stated effect is affecting calling object, 0 otherwise.

***Note:** You must state the numerical spelleffect ID, not a spell ID. These can be found listed in [Appendix B](#).*

GetForceJump (Tribunal)

GetForceJump [short value returned]

This will return a value of 1 if calling actor is affected by the [ForceJump](#) function, 0 otherwise.

GetForceMoveJump (Tribunal)

GetForceMoveJump [short value returned]

This will return a value of 1 if calling actor is affected by the [ForceMoveJump](#) function, 0 otherwise.

GetForceRun (Tribunal)

GetForceRun [short value returned]

This will return a value of 1 if calling actor is affected by the [ForceRun](#) function, 0 otherwise.

GetForceSneak

GetForceSneak [short value returned]

This will return a value of 1 if calling actor is affected by the [ForceSneak](#) function, 0 otherwise.

GetHealthGetRatio

GetHealthGetRatio [float value returned]

This will return a value of the current actor/PC health as a percentage (in decimal form) of their maximum health.

***Note:** The TESCS states to use `GetHealthRatio` to determine this value, however that function is incorrect, use this instead.*

GetInterior

GetInterior [short value returned]

This will return a value of 1 if PC is in an interior cell. 0 if an exterior cell.

GetItemCount

GetItemCount object_ID [short value returned]

This will return a value of the amount of the stated object within calling object.

***Note:** There is an issue when used with containers apparently. If the PC chooses the 'Take All' option and empties out a container, this function will return a value of the original amount placed in the container with the CS, not the current value of 0.*

GetJournalIndex

GetJournalIndex journalindex_ID [short value returned]

This will return the *last* journal entry made for the stated journal ID (not necessarily the highest journal entry). Use this to keep track how far along a quest the PC is.

GetLineOfSight

GetLineOfSight object_ID [short value returned]

This will return a value of 1 if calling object has a clear line of sight to the stated object. May not work correctly if used on non-actor objects.

GetLocked

GetLocked [short value returned]

This will return a value of 1 if object is locked, 0 otherwise.

GetLOS

GetLOS [float value returned]

Same as `GetLineOfSight` above.

GetMasserPhase

GetMasserPhase [short value returned]

This will return a value stating the phase the moon Masser is in. Apparently when a game is loaded both moons start as full. Also, when the PC enters an interior area, this function will keep returning the same result from when last in an exterior cell, until another exterior cell is entered. The value returned:

- 0 = New moon or Interior Cell
- 1 = Waxing or Waning Cresecent
- 2 = Waxing or Waning Half
- 3 = Waxing or Waning Gibbous
- 4 = Full Moon

GetPCCell

GetPCCell cell_ID [short value returned]

This will return a value of 1 if PC is in stated cell, 0 otherwise. This will return a value of 1 if the stated cell is even a subcell (i.e. a test for the cell Vivec will return a value of 1 for any cell's name that starts with 'Vivec', to include Vivec, Foreign Quarter or Vivec, Redoran Quarter, etc).

GetPCInJail (Bloodmoon)

GetPCInJail [short value returned]

This will return a value of 1 if the PC is in jail or if traveling, 0 otherwise.

GetPCJumping (Tribunal)

GetPCJumping [short value returned]

This will return a value of 1 if the PC jumps, 0 otherwise.

GetPCRank

GetPCRank faction_ID [float value returned]

This will return a value equal to the stated faction rank of the PC. If no faction is stated the calling actor's faction is used. This will return a value of -1 if the PC is not of the stated faction.

GetPCRunning (Tribunal)

GetPCRunning [short value returned]

This will return a value of 1 if the PC runs, 0 otherwise.

GetPCSleep

GetPCSleep [float value returned]

This will return a value of 1 if the PC is sleeping, 0 otherwise.

GetPCSneaking (Tribunal)

GetPCSneaking [short value returned]

This will return a value of 1 if the PC sneaks, 0 otherwise.

GetPCTraveling (Bloodmoon)

GetPCTraveling [short value returned]

This will return a value of 1 if the PC is fast traveling (boat or silt strider), 0 otherwise.

GetPlayerControlsDisabled

GetPlayerFightingDisabled

GetPlayerJumpingDisabled

GetPlayerMagicDisabled

GetPlayerLookingDisabled

GetPlayerViewSwitch (BROKEN)

GetVanityModeDisabled

GetPlayerControlsDisabled [float value returned]
GetPlayerFightingDisabled [float value returned]
GetPlayerJumpingDisabled [float value returned]
GetPlayerMagicDisabled [float value returned]
GetPlayerLookingDisabled [float value returned]
GetVanityModeDisabled [float value returned]

This will return a value of 1 if the stated function is disabled, 0 otherwise.

***Note:** The function `GetPlayerViewSwitch` is broken, use `GetVanityMode` instead.*

GetPos

GetPos axis_ID [float value returned]

This will return a value of the coordinate upon the stated axis a calling object exists. Will return a value of 0 if the object is not in the same cell as the PC.

GetRace

GetRace race_ID [short value returned]

This will return a value of 1 if calling actor/PC is of the stated race, 0 otherwise.

GetSecondsPassed

GetSecondsPassed [float value returned]

This will return the current amount of time, or seconds, that have passed since the last frame. A common and useful way to keep track of time in a script is to type the following:

```
Set [variable] to ( GetSecondsPassed + [variable] )
```

GetSecundaPhase

GetSecundaPhase [short value returned]

This will return a value stating the phase the moon Secundus is in. Apparently when a game is loaded both moons start as full. Also, when the PC enters an interior area, this function

will keep returning the same result from when last in an exterior cell, until another exterior cell is entered. The value returned:

- 0 = New moon or Interior Cell
- 1 = Waxing or Waning Cresecent
- 2 = Waxing or Waning Half
- 3 = Waxing or Waning Gibbous
- 4 = Full Moon

***Note:** In the TESCS help file this function is incorrectly stated as `GetSecundusPhase`.*

GetSoundPlaying

`GetSoundPlaying sound_ID [short value returned]`

This will return a value of 1 if stated sound is playing, 0 otherwise.

GetSpell

`GetSpell spell_ID [short value returned]`

This will return a value of 1 if the stated spell is listed in actor/PC's spell inventory, 0 otherwise. This will only work for 'spells', not abilities, powers, curses, or diseases.

GetSpellEffects

`GetSpellEffects spell_ID [short value returned]`

This will return a value of 1 if the calling object is being affected by the stated spell effect, 0 otherwise. This will work with spells, abilities, powers, curses, and diseases.

GetSpellReadied (Tribunal)

`GetSpellReadied [short value returned]`

This will return a value of 1 if the calling actor/PC has a spell readied to be cast (has their hands up ready), 0 otherwise.

GetSquareRoot (Tribunal)

`GetSquareRoot [variable]`

This will compute the square root of the stated variable.

GetStandingActor

`GetStandingActor [short value returned]`

This will return a value of 1 if an actor or the PC is standing on the calling object, 0 otherwise.

GetStandingPC

`GetStandingPC [short value returned]`

This will return a value of 1 if the PC is standing on the calling object, 0 otherwise.

GetStartingAngle

`GetStartingAngle axis_ID [float value returned]`

This will return the value of the starting coordinate along the stated axis of the calling object.

GetStartingPos

`GetStartingPos axis_ID [float value returned]`

This will return the value of the starting grid (in degrees) along the stated axis of the calling object.

GetSuperJump

`GetSuperJump [short value returned]`

This will return a value of 1 if this has been activated with `SetSuperJump == 1`, 0 otherwise. I activated this ingame to see what happened with my PC. Nothing, I could detect no change in how my PC jumped or moved. I have no idea what this would be for.

GetTarget

`GetTarget actor_ID [short value returned]`

This will return a value of 1 if the calling object is targeting the stated actor, 0 otherwise.

GetWaterBreathing

`GetWaterBreathing [short value returned]`

This will return a value of 1 if actor/PC is able to breathe water, 0 otherwise.

GetWaterWalking

`GetWaterWalking [short value returned]`

This will return a value of 1 if actor/PC is able to walk on water, 0 otherwise.

GetWeaponDrawn (Tribunal)

`GetWeaponDrawn [short value returned]`

This will return a value of 1 if the calling actor has a weapon drawn or has fists ready for combat, 0 otherwise.

GetWeaponType (Tribunal)

`GetWeaponType [short value returned]`

This will return a value of the current equipped weapon as stated below (whether or not drawn):

- 1 = Unarmed
- 0 = Short blade, 1-Hand
- 1 = Long blade, 1-Hand
- 2 = Long blade, 2-Hand
- 3 = Blunt, 1-Hand
- 4 = Blunt, 2-Hand Close

5 = Blunt, 2-Hand Wide
6 = Spear, 2-Hand Wide
7 = Axe, 1-Hand
8 = Axe, 2-Hand Close
9 = Bow
10 = Crossbow
11 = Thrown Weapon
12 = Arrow
13 = Bolt

GetWerewolfKills (Bloodmoon)

`GetWerewolfKills` [short value returned]

This will return a value of the number kills the PC last made as a werewolf.

GetWindSpeed

`GetWindSpeed` [float value returned]

This will return a value of the current wind speed. This number fluctuates constantly, but generally sits somewhere between 0 and 2 for a normal calm day.

GoToJail

`GoToJail`

This will send the PC instantly to jail.

HasItemEquipped (Tribunal)

`HasItemEquipped` object_ID [short value returned]

This will return a value of 1 if the stated object is equipped on the actor/PC, 0 otherwise.

HasSoulGem

`HasSoulGem` creature_ID [short value returned]

This will return a value of 1 if the calling object has a soulgem with the stated creature within it's inventory, 0 otherwise.

HitAttemptOnMe

`HitAttemptOnMe` object_ID [short value returned]

This will return a value of 1 for one frame if an attempt is made to strike calling object with stated object, 0 while no attempt is made.

HitOnMe

`HitOnMe` object_ID [short value returned]

This will return a value of 1 for one frame if a successful strike is made on calling object with stated object, 0 otherwise.

HurtCollidingActor (Tribunal)

HurtCollidingActor [damage]

This will apply the stated damage value to any actor or the PC that collides with the calling object.

HurtStandingActor

HurtStandingActor [damage]

This will apply the stated damage value to any actor or the PC that is standing on the calling object. If the damage number is a negative value, the actor/PC takes damage, if a positive number they will be healed.

IsWerewolf (Bloodmoon)

IsWerewolf [short value returned]

This will return a value of 1 if calling actor or PC is in werewolf form, 0 otherwise.

Journal

Journal name_ID, entry_number

This is used to add a new entry to the journal when the PC attains some point in a quest (or just for info purposes, or whatever). The name ID is the unique name ID that the entry falls under, and the number it is placed in the journal (consecutive entries should always have higher numbers as they are placed).

Lock

Lock lock_level

This will lock the calling object to the stated lock level.

LoopGroup

LoopGroup groupname_ID, [number], [flag]

This will make the stated groupname (i.e. animation) play the stated 'number' of times consecutively once the current animation is done. Thus you could get a specific idle animation to play repeatedly. The flag is optional but will:

- 0 = Current animation will finish all cycles and then this will play
- 1 = Current animation will immediately stop and this animation will start to play from it's beginning
- 2 = Current animation will immediately stop and this animation will start to play from the beginning of it's loop cycle.

MenuMode

MenuMode [short value returned]

This will return a value of 1 if PC is in menu mode, 0 otherwise. Sleep mode and the level up menu will also return a value of 1 as they are considered being in menu mode too. Include it by typing the following at the top of all other code, like this:

```

if ( MenuMode == 1 )
    Return
endif

```

Note: This is an important function. You should put this at the top of almost every script you write. What it does is stop a script from running while the player has their menus open, thus the script is on hold until they close their menus. This prevents the PC from being unable to react to unpleasantness by opening the menus, or somehow benefiting by being sly enough to do so. It also helps prevent a lagging arrow (we all know about this). If this doesn't make sense, don't worry, just include it.

MessageBox

```

MessageBox "text"
MessageBox "blah blah blah text, [variable] blah blah", [variable], "button1", "button2"

```

This function gives you the ability to display text ingame. It appears bottom middle of the screen for the player to read. The text will float for several seconds and then disappear. You can include a button that will keep the text displayed until it's clicked, or you can include up to five buttons, each being a different choice to a question you ask the player. You can also introduce variables for display in text, including certain specific ones. Whew! A lot you can do with this, so let's look at it carefully.

To simply get some text displayed you want the player to read, include the function in a script with the text typed after it in quotes. Thus:

```

MessageBox "put your text here"

```

Let's say you want text to be displayed and stay on the screen until the player acknowledges it. You have to include a button to do this. A button is added by putting a comma after the quoted text, and then put **Ok** in quotes. Thus:

```

MessageBox "make player read this", "Ok"

```

Note: It doesn't matter what word you use. "Ok" just makes sense (to me). Nor does only having one button make a difference. In the script if you don't assign an action to a button, nothing happens after the button gets pressed, the message box just goes away.

Ok, now, let's say you ask the player a question of some sort and you need them to respond with a choice. You want to offer them several choices (up to 5, that's the limit per message box. Want to ask more? Then split the questioning up among several message boxes). Same format as above, just include a comma to separate each stated button. Thus:

```

MessageBox "Question", "button1", "button2", "button3", "button4"

```

OR

```

MessageBox "How do you want to die Wimp!", "Rubber Chicken", "Wet Noodle", "Blighted ex-mother in law", "Veronica Zeminova wearing only jello"

```

Ok, now you got buttons. What happens when the player clicks one? Oh yeah... you almost forgot that part (stop thinking of jello and focus here). You must introduce the button choice results with the `GetButtonPressed` function. Keep in mind: You must name your button under the `GetButtonPressed` function the same as under the `MessageBox` function. This can be a very useful function for this choose-a-button ability alone. But wait! There's more!

You can also introduce variables in text. The type of variable you wish to include determines the syntax you use:

Float variables use: `%.2f`

Short variables use: `%g`

Long variables use: `%g`

Special variables use: `^variable_name`

Ok, stay with me here. To use a float variable you just include `%.[number]f` in the text somewhere, and then a comma, then state the variable after that. The variable must be a variable from within the script, you can't use one from another script. The percent symbol (%) tells the script to look for a variable, the 'number' tells the text how many decimal places to include when listing the variable (not sure if the number will get rounded off). Don't forget the decimal point. This is only for float variables. So:

```
MessageBox "I see you have brought me %.2f ounces of jello...", ounces_variable
```

To include a short or long variable you only need `%g`. Again the % tells the script to include a variable stated at the end of the function.

```
MessageBox "Why yes, I do expect all %g of you to eat all %g pounds of chicken.",  
number_of_guys_variable, pounds_variable
```

Special variables are mostly string variables (which means can have/has text not just a number), though some are globals (and you can use the above syntax to list them too). These just use the ^ symbol and then the name, you must still name the variable again at the end of the function line. The variables you can use this with are:

<code>^PCName</code>	The player's name.
<code>^PCClass</code>	The player's class.
<code>^PCRace</code>	The player's race.
<code>^PCRank</code>	The player's rank in the speaker's faction.
<code>^NextPCRank</code>	The player's next rank in the speaker's faction.
<code>^Cell</code>	The cell the player is currently in.
<code>^Global</code>	Displays the numerical value of stated global.
<code>^NPC.Name</code>	The NPC's name.
<code>^NPC.Race</code>	The NPC's race.
<code>^NPC.Class</code>	The NPC's class.
<code>^NPC.Faction</code>	The NPC's faction. If they have no faction, it will be blank.
<code>^NPC.Rank</code>	The NPC's rank.

```
MessageBox "Aha! Now I have you ^PCName, you may have stolen all %g pounds of chicken!  
But I still have over %.3f ounces of jello! And you'll never get it or my name isn't  
^NPC.Name!", pounds_variable, ounces_variable
```

Keep in Mind: You must list any variables used after the text quote, in the same order that you used them in the text, and you must separate them by a comma and a space. This is how the game knows what variable to include where. If you use a variable more than once, you must list it, in order, more than once.

Mod-

```
ModEndurance [amount]
ModArmorer [amount]
ModCastPenalty [amount]
[float value returned]
```

Most, but not all, functions that can use `Get-/Set-`, you can also use `Mod-` with (i.e. `ModStrength`, `ModBlock`, `ModFlee`, etc.). There are some exceptions. `Mod-` is like the `Set-` function in that it can be used to increase or decrease a stat for an actor or PC. But whereas with `Set-` you can change a stat to any numerical value the game will accept, even those above the normal maximum value, with `Mod-` you can only change a stat to a numerical value that is within the normal minimum/maximum range for that actor/PC. This means for most stats you can only modify or change the stat within a range of 1-100 (to get a stat value over 100 you would have to use the `Set-` function). I list below the viable functions. Refer to the `Get-` function listing to see a description of any of these functions. I will list those functions that have a syntax different than under `Get-`.

Agility	Acrobatics	ArmorBonus
Alarm	Alchemy	AttackBonus
Disposition	Alteration	Blindness
Endurance	Armorer	CastPenalty
Fatigue	Athletics	Chameleon
Fight	Axe	FactionReaction
Flee	Block	Flying
Health	BluntWeapon	Invisible
Hello	Conjuration	Paralysis
Intelligence	Destruction	PCCrimeLevel
Level	Enchant	PCFacRep
Luck	HandToHand	PCVisionBonus
Magicka	HeavyArmor	ResistBlight
Personality	Illusion	ResistCorprus
Reputation	LightArmor	ResistDisease
Speed	LongBlade	ResistFire
Strength	Marksman	ResistFrost
Willpower	MediumArmor	ResistMagicka
	Mercantile	ResistNormalWeapons
	Mysticism	ResistParalysis
	Restoration	ResistPoison
	Security	ResistShock
	ShortBlade	Scale
	Sneak	Silence

Spear	SuperJump
Speechcraft	Swim Speed
Unarmored	WaterBreathing
	WaterLevel
	WaterWalking

Note: *ModFatigue, ModHealth, and ModMagicka will change the maximum value of the stat's base up to the normal possible maximum. To change the current value you must use ModCurrent- below.*

ModCurrentFatigue**ModCurrentHealth****ModCurrentMagicka**

[float value returned]

Each of these functions will modify or change the current value up to the current maximum for the actor/PC.

ModRegion

ModRegion, RegionID, [Clear], [Cloudy], [Foggy], [Overcast], [Rain], [Thunder], [Ash], [Blight], [Snow?], [Blizzard?]

This function is used to change the percentage chance of each weather type occurring. The last two; snow and blizzard are not verified. All assigned values must equal 100.

Move

Move axis_ID, [speed]

This will move or rotate an object upon itself. It will not change grid coordinates in the world. You can only rotate on one axis at a time, and you must state the speed at which it will rotate. Be aware that the speed an object rotates is affected by the ability of the computer system the game is running on. Thus if you need an object to rotate a given amount within a certain amount of time, you should set a condition to complete the action within the specified set amount of time.

MoveWorld

MoveWorld axis_ID, [speed]

This will move or rotate an object upon the world grid. You can only move along one axis at a time. Be aware that the speed an object moves is affected by the ability of the computer system the game is running on. Thus if you need an object to move a given amount within a certain amount of time, you should set a condition to complete the action within the specified set amount of time.

OnActivate

OnActivate

This will return a value of 1 for one frame if the calling object is activated, 0 otherwise. Use this to determine when/if the PC activates an object, and then have said object do

something else. If you use this function you must then use [Activate](#) if you want the object to normally activate (as it normally would).

OnDeath
OnDeath

This will return a value of 1 for one frame if the calling actor dies, 0 the rest of the time.

OnKnockout
OnKnockout

This will return a value of 1 for one frame if the calling actor is knocked out, 0 the rest of the time.

OnMurder
OnMurder

This will return a value of 1 for one frame if the calling actor is killed, 0 the rest of the time.

***Note:** OnMurder and OnDeath will both return a value of 1 for one frame if used in the same script if an actor is killed. I do not suggest using both functions in the same script for the same actor.*

PayFine
PayFine

When called will remove gold from the PC's inventory equal to the bounty on their head for crimes committed, remove all stolen items and place them in the nearest stolen item chests around the island, and clear the PC of all crimes. This is the function used by the game when the guards catch the PC after committing a crime.

PayFineThief
PayFineThief

When called this will remove gold from the PC's inventory equal to half the bounty on their head for crimes committed, and clear the PC of all crimes. This will not remove any stolen items in the PC's inventory. This is the function used by the game when the PC uses the Thieve's Guild to clear bounties from their head due to crimes they've committed.

PCClearExpelled
PCClearExpelled function_ID

If the PC has been expelled from the stated faction, use this function to clear that flag and allow the PC to reenter the faction.

PCExpell
PCExpell faction_ID

This will expel the PC from the stated faction.

PCExpelled

PCExpelled faction_ID

Returns a value of 1 if the PC has been expelled from stated faction, 0 otherwise.

PCForce3rdPerson

PCForce3rdPerson

This will force the PC into 3rd person view. The change is put in the queue to change when the current animation is finished.

PCForce1stPerson

PCForce1stPerson

This will force the PC into 1st person view. The change is put in the queue to change when the current animation is finished.

PCGet3rdPerson

PCGet3rdPerson

This will return a value of 1 if the PC is in 3rd person view, 0 otherwise.

PCJoinFaction

PCJoinFaction faction_ID

When called the PC will 'join' the stated faction. If no faction is stated, the PC will join the calling actor's faction.

PCLowerRank

PCLowerRank faction_ID

This will lower the PC a rank in the stated faction.

PCRaiseRank

PCRaiseRank faction_ID

This will raise the PC a rank in the stated faction.

PlaceAtMe

PlaceAtMe object_ID, [number], [distance], [position]

This will place the specified number of stated objects, the defined distance in units from the calling actor/PC, in one of four positions around the actor/PC using a value listed below:

0 = front

1 = back

2 = left

3 = right

PlaceAtPC

PlaceAtPC object_ID, [number], [distance], [position]

This will place the specified number of stated objects, the defined distance in units from the calling PC, in one of four positions around the PC using a value listed below:

- 0 = front
- 1 = back
- 2 = left
- 3 = right

PlaceItem (Tribunal)

PlaceItem object_ID, x, y, z, [facing]

This will place the stated object at the x, y, z coordinates, front towards the direction stated in the facing value (stated in degrees), in an exterior cell. This function can be used to place an object in the world that had not been placed there with the CS.

PlaceItemCell (Tribunal)

PlaceItemCell object_ID, cell_ID, x, y, z, [facing]

This will place the stated object at the x, y, z coordinates, front towards the direction stated in the facing value (stated in degrees), in a stated interior cell. This function can be used to place an object in the world that had not been placed there with the CS.

***Note:** If you use this function, any object placed will appear as you stipulate. However, it's been reported that when the game is saved, exited, then loaded, it will not be present. Very odd, haven't messed with this function yet.*

PlayBink

PlayBink "filename", [flag]

This will play the specified filename in the bink format. If the flag is set to 1 the player can skip the movie by pressing the 'esc' key, if 0 they cannot.

PlayGroup

PlayGroup groupname_ID, [flag]

This will play the specified animation groupname. The flag will determine where in the queue it goes. The flag values are:

- 0 = Current animation will finish all cycles and then this will play
- 1 = Current animation will immediately stop and this animation will start to play from it's beginning
- 2 = Current animation will immediately stop and this animation will start to play from the beginning of it's loop cycle.

PlayLoopSound3D

PlayLoopSound3D, "sound_ID"

This will play the specified sound as if coming from the calling object, thus as the PC moves closer/farther away the sound gets louder/dimmer. This will play the sound until the [StopSound](#) function is used.

PlayLoopSound3DVP

```
PlayLoopSound3DVP, "sound_ID", [volume], [pitch]
```

This will play the specified sound as if coming from the calling object, thus as the PC moves closer/farther away the sound gets louder/dimmer. The value for volume can be set to 1 (full), or 0 (none). Pitch also can only be set to 1 or 0. This will play the sound until the [StopSound](#) function is used.

PlaySound

```
PlaySound, "sound_ID"
```

This will play the specified sound file.

PlaySound3D

```
PlaySound3D, "sound_ID"
```

This will play the specified sound as if coming from the calling object, thus as the PC moves closer/farther away the sound gets louder/dimmer.

PlaySound3DVP

```
PlaySound3DVP, "sound_ID", [volume], [pitch]
```

This will play the specified sound as if coming from the calling object, thus as the PC moves closer/farther away the sound is gets louder/dimmer. The value for volume can be set to 1 (full), or 0 (none). Pitch also can only be set to 1 or 0.

PlaySoundVP

```
PlaySoundVP, "sound_ID", [volume], [pitch]
```

This will play the specified sound file. The value for volume can be set to 1 (full), or 0 (none). Pitch also can only be set to 1 or 0.

Position

```
Position object_ID, x, y, z, [facing]
```

This will place the stated object at the x, y, z coordinates, front towards the direction stated in the facing value (stated in degrees), in an exterior cell. If the PC moved and there is clipping they will be moved to the closest open spot. If you move an object with a script attached, the script will not process until the cell is exited and then reloaded by the game. Apparently, the facing value doesn't seem to work on NPCs.

PositionCell

```
PositionCell object_ID, cell_ID, x, y, z, [facing]
```

This works the same as the [Position](#) function, except you use it with interior cells, hence why you must state the cell ID. If you put an exterior cell ID in the function it will work with

exterior cells. This function doesn't seem to have the problems that the Position function sometimes has.

RaiseRank

RaiseRank

Raises the calling object one rank in its faction.

RemoveEffects

RemoveEffects effects_numerical_ID

This will remove any spells affecting the calling actor/PC that include the stated effect ID.

Note that you cannot use the spell effect name ID, you must use the numerical value assigned to each spell effect by the game. This will not remove just the stated spell effect, it will remove any spell the spell effect is a part of. These are listed in [Appendix B](#).

RemoveFromLevCreature (Tribunal)

RemoveFromLevCreature levcreaturelist_ID, creature_ID, [level]

This will remove a creature or creature list from a leveled list, assuming you get the parameters correct (or it won't remove it). The leveled creature list ID is the list to remove the creature from. The creature ID is the creature (or creature list ID) to remove, and the level value is the listed level in the list the creature appears at.

RemoveFromLevItem (Tribunal)

RemoveFromLevItem levitemlist_ID, item_ID, [level]

This will remove an item or item list from a leveled list, assuming you get the parameters correct (or it won't remove it). The leveled item list ID is the list to remove the creature from. The item ID is the item (or item list ID) to remove, and the level value is the listed level in the list the item appears at.

RemoveItem

RemoveItem object_ID, [amount]

This will remove the stated object from the calling object's inventory. If you remove an object that does not exist in the inventory, or more than is present, the full amount of items are not removed, but the encumbrance value is changed incorrectly. Thus a naked PC would still be encumbered. To prevent this always use [GetItemCount](#) to check for the number of any items before you try to remove them.

***Note:** You cannot use this function in a script attached to an object you want to remove, if you do, you will crash the game. Instead use a different or global script safely.*

RemoveSoulGem

RemoveSoulGem creature_ID

This will remove one soulgem that contains the stated creature ID.

RemoveSpell

`RemoveSpell spell_ID`

This will remove the stated spell from the calling actor/PC's spell inventory. If the 'spell' is a curse or a disease affecting the actor/PC, it will be removed.

RemoveSpellEffects

`RemoveSpellEffects spell_ID`

This will remove the stated spell ID from the list of effects affecting the calling actor/PC.

Note that you use the spell name ID, not the spell effect numerical ID as in [Remove Effects](#).

RepairedOnMe

`RepairedOnMe repairobject_ID`

This will return a value of 1 if the stated repair object is used to repair the calling object.

Resurrect

`Resurrect`

This will bring an actor back to life by resetting them to their starting point and original configuration as set in the CS. Reports state possible game crashes when used.

Return

`Return`

This function tells the script to stop processing this frame and start over the next frame. This is a useful function. Use this to keep a script from processing in its entirety every frame if unnecessary. Refer to [MenuMode](#) for an example.

Rotate

`Rotate axis_ID, [angle]`

This will rotate the calling object along it's own stated axis, the specified number of degrees per second.

RotateWorld

`RotateWorld axis_ID, [angle]`

This will rotate the calling object along the world's stated axis, the specified number of degrees per second.

SameFaction

`SameFaction`

This will return a value of 1 if the PC is in the same faction as the calling actor, 0 otherwise.

Say

`Say "filename_ID", "text"`

[Table of Contents](#)

[Table of Contents](#)

This will make the actor 'say' the stated filename, the specified text will be displayed as they speak (these are usually the same words, but don't have to be). Only works with actors.

SayDone

SayDone

This will return a value of 1 if the object is *not* speaking, 0 if they are speaking. Useful for checking if they have finished 'saying' something.

ScriptRunning

ScriptRunning script_ID

This will return a value of 1 if the stated script is currently processing, 0 otherwise.

Set-

SetLevel [amount]
SetEnchant [amount]
SetPCFacRep [amount]
Etc.

The Set- function is used to change a stat (or game aspect) to a new value. This value can be above the normal maximum value for that stat or aspect. Refer to [Get-/Mod-](#) for further info on these functions. Be advised not all functions that can be used as Get- or Set- can be used with Mod-.

Agility	Acrobatics	ArmorBonus
Alarm	Alchemy	AttackBonus
Disposition	Alteration	Blindness
Endurance	Armorer	CastPenalty
Fatigue	Athletics	Chameleon
Fight	Axe	FactionReaction
Flee	Block	Flying
Health	BluntWeapon	Invisible
Hello	Conjuration	Paralysis
Intelligence	Destruction	PCCrimeLevel
Level	Enchant	PCFacRep
Luck	HandToHand	PCVisionBonus
Magicka	HeavyArmor	ResistBlight
Personality	Illusion	ResistCorprus
Reputation	LightArmor	ResistDisease
Speed	LongBlade	ResistFire
Strength	Marksman	ResistFrost
Willpower	MediumArmor	ResistMagicka
	Mercantile	ResistNormalWeapons
	Mysticism	ResistParalysis
	Restoration	ResistPoison
	Security	ResistShock
	ShortBlade	Scale
	Sneak	Silence

Spear	SuperJump
Speechcraft	Swim Speed
Unarmored	WaterBreathing
	WaterLevel
	WaterWalking

SetAngle

```
SetAngle axis_ID, [angle]
```

This will set an object to face the specified direction as determined by angle (in degrees), along the stated axis.

SetAtStart

```
SetAtStart
```

This will reset an object to its original starting grid position. I have come across statements saying this does not always function exactly right.

SetDelete (Tribunal)

```
SetDelete, [flag]
```

This is used to completely delete an object from the game. There are some considerations to keep in mind if you truly need to delete an object. Make sure it will never be needed or should not ever be used again. If you decide to delete an object, [Disable](#) it first in a script, then wait a few frames before you delete it. I suggest giving a script roughly 10 frames in between. The flag is used to mark either delete object (use a value of 1) or if an object is deleted you might be able to undo it (use a value of 0, haven't verified this though). Basically, if an item is in the game from a plug-in or a master file, the deletion is not permanent until the game is saved. If the item is added ingame with a script or console function (such as [AddItem](#), or [PlaceAtPC](#)), the item is immediately deleted and cannot be undone. In any script you use `SetDelete` you should have something like this in it:

```
Short timer

Set timer to ( timer + GetSecondsPassed )
if ( timer == 0 )
    object_ID->Disable
endif

if ( timer < 10 )
    Return
endif

if ( timer == 10 )
    object_ID->SetDelete, 1
endif
```

Note: If you use either [ExplodeSpell](#) or [Cast](#) in a script with an object, wait several seconds for the effects to finish before you disable then delete the object(s).

SetJournalIndex

```
SetJournalIndex, journal_ID, [index number]
```

This will set a stated journal ID to a specified index number. This means effectively you can change a journal indexed already to a new or different number. If you set the index higher than the highest index listed for that Journal entry, when the game is next loaded to play, the index will reset to the actual highest index (useful to check if a player has reloaded a game).

SetPos

SetPos axis_ID, [grid number]

This will set the calling object to the specified grid number along the stated axis. This will only work within the current cell. If in an exterior this will only work for objects within loaded cells (i.e. current and 2 cell buffer).

SetWerewolfAcrobatics (Bloodmoon)

SetWerewolfAcrobatics [stat value]

This will set the actor's skills to werewolf form, as defined by fWereWolfXXXX GMST values.

ShowMap

ShowMap, cell_ID

This will remove the 'fog' covering a cell on the player's map if they haven't visited it, or if only partially explored. It will also display all named areas on the map. Note that this will affect all cells that begin with the stated cell ID name. Thus if you use a cell ID of 'Vivec', any cell that begins with Vivec, to include such as 'Vivec, Foreign Quarter', will be affected.

SkipAnim

SkipAnim

This function will prevent any animation from being played during the frame it is processed.

StartCombat

StartCombat actor_ID

This will cause the calling actor to begin combat with the stated object/actor, or the PC.

StartScript

StartScript, script_ID

This will start the stated script. Note that you cannot start a script from within itself (like duh), but must be started from a separate local or global script. I suggest using a local script attached to something not noticeable, and consider disabling and or deleting the object (if not needed or would be out of place).

StopCombat

StopCombat

This can be used to make an actor stop attacking another actor or the PC.

StopScript

StopScript, script_ID

This is used to stop a script from being processed. You *must* use this in a different script than the one you want to stop. When this function is used in a script, the script will still function and process until the End command is reached. So make sure if you use this, no further functions or commands will execute.

StopSound

StopSound sound_ID

This will stop the stated sound if playing.

StreamMusic

StreamMusic "filename_ID"

This will play the stated filename.

TurnMoonRed (Bloodmoon)

TurnMoonRed

This will turn the moons red. This is for specific effects associated with werewolf quests and scripts.

TurnMoonWhite (Bloodmoon)

TurnMoonWhite

This will turn the moons white. This is for specific effects associated with werewolf quests and scripts.

UndoWerewolf (Bloodmoon)

UndoWerewolf

This will change an actor/PC from werewolf form to their normal form. Note that Bloodmoon quests depend on this function so using it without consideration might cause problems.

Unlock

Unlock

This will unlock the calling object.

UsedOnMe (BROKEN)

Probably originally to be used to test if an object is used on another object. Don't use this, it won't work and might even cause problems if you try.

WakeUpPC
WakeUpPC

If the PC is sleeping this will wake them up.

Global and Hard Coded Variables. There are quite a few hard coded variables that can be used in the game. The global variables you don't need to declare, as they already exist and hold data for you to access. Others are hard coded in how they work, but you must declare them, as they hold no data till you assign it to them. All these variables can be used as any other variable. I will list them below and give a description of each.

AllowWereWolfForceGreeting
AllowWereWolfForceGreeting (short global)

You must declare this variable, but do not have to reference it in a script, merely declare it. This allows the use of the `ForceGreeting` function on the PC when in werewolf form.

Day
Day (short global)

You do not have to declare this variable. It holds the current day of the month in the game.

DaysPassed
DaysPassed (short global)

You do not have to declare the variable. It holds the number of days since the current game was begun.

GameHour
GameHour (float global)

You do not have to declare this variable. It holds the current time of day (game time).

Month
Month (short global)

You do not have to declare this variable. It holds the current month of the year. Each month is numbered 0-11. There is a bug that when the year ends the game goes from month 11 to 1, skipping the first month of the year. I have not been able to determine if it is fixed with any patches.

OnPCAdd
OnPCAdd [short value returned]

You must declare this variable to use it. Use this within a script called by an object. If the object is added to the PC's inventory it will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnPCDrop

OnPCDrop [short value returned]

You must declare this variable to use it. Use this within a script called by an object. If the object is dropped from the PC's inventory it will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnPCEquip

OnPCEquip [short value returned]

You must declare this variable to use it. Use this within a script called by an object. If the PC equips the object, this will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnPCHitMe

OnPCHitMe [short value returned]

You must declare this variable to use it. Use this within a script called by an object. If the PC strikes said object this will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnPCRepair

OnPCRepair [short value returned]

You must declare this variable to use it. Use this within a script called by an object. If the PC repairs the object this will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnPCSoulGemUse

OnPCSoulGemUse [short value returned]

You must declare this variable to use it. Use this within a script to check if the PC uses any soulgem to enchant or recharge an item. If so, this will return a value of 1, otherwise it will return a value of 0. To use the variable in the script again you must change the value back to 0 before using it.

OnRepair (BROKEN)

OnRepair

This variable does not work. It was supposed to return a value of 1 if an attempt is made to repair a calling object.

PCKnownWerewolf

PCKnownWerewolf (short global)

This returns a value of 1 if the PC is known to be a werewolf, 0 otherwise

PCRace

PCRace (short global)

You do not have to declare this variable. This holds a value assigned by the PC's race. If you create a new race and want it to have a value also, just open up the script named 'PCRace' (Whoa! Who would've guessed!?), and add your race accordingly and give it a value of 11 (or greater if more than one race is added). The value assigned:

- 1 = Agonian
- 2 = Breton
- 3 = Dark Elf
- 4 = High Elf
- 5 = Imperial
- 6 = Khajiit
- 7 = Nord
- 8 = Orc
- 9 = Redguard
- 10 = Woodelf

***Note:** If a player opens the console and changes their race, this value will not be changed, they will be identified as their prior race.*

PCSkipEquip

PCSkipEquip (short global)

You must declare this variable to use it. If set to 1 and a called object is activated it cannot be equipped, 0 allows equipping. This function has issues. For some reason, when a book is equipped (read) from inventory, this function returns a value of 1 (instead of [OnPCEquip](#)). And under certain circumstances it may produce other odd results, though most of the time it works fine.

PCVampire

PCVampire (short global)

You do not have to declare this variable. This holds a value stating the PC's status as a vampire. The value indicates:

- 1 = cured of vampirism (immune to vampirism)
- 0 = not a vampire
- 1 = currently a vampire

PCWerewolf

PCWerewolf (short global)

You do not have to declare this variable. This holds a value stating the PC's status as a werewolf. The value indicates:

- 1 = cured of lycanthropy (immune to lycanthropy)
- 0 = not a werewolf
- 1 = currently a werewolf

StayOutside

StayOutside (short global)

You must declare this variable to use it. When set to 1 in a script this will make an actor stay 'outside' in an exterior cell if the PC enters an interior cell. They will wait where the PC left them, and upon meeting up with them will rejoin with the PC.

Year

Year (short global)

You do not have to declare this variable. This will return the current game year.

Character Generation. So you want to know how to get the game to start the player off somewhere besides the prison ship, and have them generate a character through some other means besides going through the Census and Exercise Office, right? This can be done. But you will have to understand that this will take many changes to the start of the game. If you aren't careful you could be setting your plug-in, or a player, up for some frustrating failures. Second, the start of the game is set up to be a one time deal. Once you go through the process the game removes it so you can't go back and start character creation over again. This is done to protect the game from problems. Third you must know what to 'undo' of this process and how to 'redo' it the way you want (do not just delete everything involved, I've mentioned before that this, in general, is a bad thing). This means you have to know how to do some simple scripting at the very least. I will go through all this in order. First let's review the functions involved.

***Note:** the PC is actually listed under the NPC tab in the Object Window. Both ID and Name are 'player' (what a surprise). The PC starts off as an average Dark Elf Acrobat, but during the PC creation process this is changed to what the PC wants. You can make changes to the PC under this tab, but I do not recommend do more than change race or class.*

DisablePlayerControls

This is used at the start of the game to prevent the player from doing anything but looking around and talking to the NPC prisoner next to him on the ship. This is to ensure the player does what he needs to do to create his PC. Next you get asked your name, race, birthsign, class, and then are asked to review your choices. None of these functions have a corresponding Disable- function. They are use once functions. These are:

EnableNameMenu
EnableRaceMenu
EnableBirthMenu
EnableClassMenu
EnableStatReviewMenu

***Note:** These should only be used once during the PC creation process, further use could crash the game.*

As the player progresses their menus are also enabled so they can access them. This is done one at a time in the game so a new player can learn about them. You could do this also, or just enable them all at some point. You can also disable/enable them at any time during the game:

```
EnableStatsMenu  
EnableMagicMenu  
EnableMapMenu  
EnableInventoryMenu
```

Finally, there are the PC actions players can take that are enabled. These are the ability to switch views, jump, etc. You can disable/enable them also at any time during the game:

```
EnablePlayerJumping  
EnablePlayerFighting  
EnablePlayerLooking  
EnablePlayerMagic  
EnableVanityMode
```

And lastly, instead of individually enabling each menu and ability of the PC, once the PC is created you could just instead use this function:

```
EnablePlayerControls
```

This will enable everything for the PC.

Now you know what functions are used and how. Scripts are used to control the actions of the actors during this process and to lead the player through the process in a clear precise path. The scripts that control the process all start with *CharGen-*. Objects/actors involved too have the CharGen- added to the ID. Look these up and make sure you understand what they are doing. If you wish to change this process I suggest either you make very small modifications to the scripts and actors used, **OR** just start the player at another location and use your own scripts.

For a new location you must do two things:

1. Write one or more scripts to disable the player controls, enable the PC creation functions, and then enable the player controls.
2. Remove all PC creation scripts/objects/areas from Seyda Neen.
3. Edit the Main Script to start the PC somewhere besides the CharGen boat.

Create your PC creation process, be diligent and get it right. Now you must make changes to the Seyda Neen cells and those actors and scripts that involve character generation. This is to prevent the PC at some point coming to the census and exercise office and accidentally triggering the process again. The simplest way to do this? Just delete the references (note I said *references*, not the actual *objects*) in the cell that are part of the process. The problem with this? You break the main quest before the PC ever even starts it (sort of). To get it going in the first place you must:

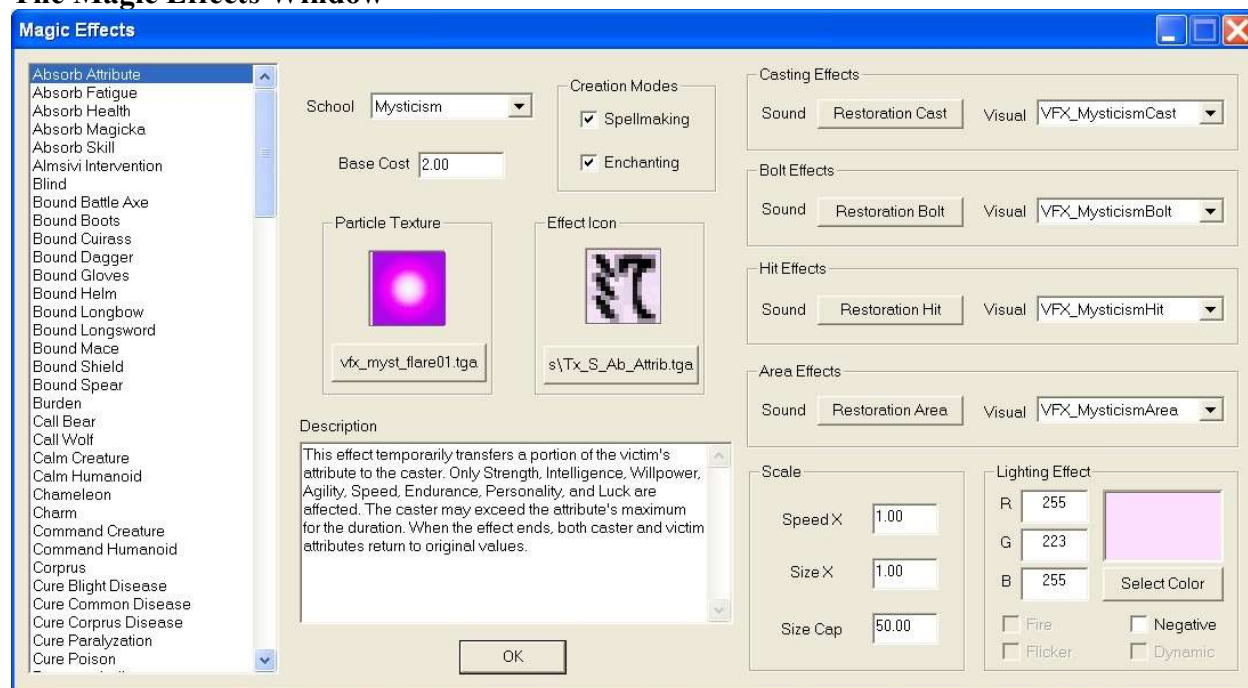
1. Give the package meant for Caius Cossades to the player.
2. Add the appropriate journal entry for the player to go see Caius.
3. Give the player incentive to go see Caius.

Note: Normally when a player starts the game, goes through the PC creation process, all the actors and the prison ship stay present in the cell until the PC leaves it. Once the PC is

far enough away (or fast travel out with the silt strider) these CharGen objects are dropped, the game knows to disable and delete them.

How you choose to start the player off if you change the process is up to you. Just be careful, make sure the changes you institute work correctly. Make sure the player cannot reenter the cell or area they start off in and restart the PC creation process again (it may be ok to reenter the area, just not start the process). Make sure you jump start the player on his way to playing the main quest (don't depend on the PC just 'bumping' into Caius one day). Personally I don't recommend fudging with this process at all, but if you just *have* to, take care.

The Magic Effects Window



This window allows you to modify the visual effects seen when spells are cast. Spell effects (not spells) are listed in the large window on the left. Highlight an effect and the appropriate visual effects are listed in boxes and drop down menus to the right. These are explained below. Take care and give consideration to any decision to modify anything in this window as it could easily unbalance gameplay far more than intended.

School. This drop down menu lists each spell effect school. Use this to change what school a spell effect is associated with (and what skill it is based on).

Base Cost. This is the cost in magicka points as a base that the game uses to calculate the cost of casting a spell. Remember this is just a base, other factors modify the cost.

Creation Modes. If the **Spellmaking** box is checked this spell effect will be included on the list of effects the PC can use to create a spell with. If the **Enchanting** box is checked this effect will be included on the list of effects the PC can use to enchant items with.

Particle Texture. This is the particle effect used with this spell effect (the visual effect when casting).

[Table of Contents](#)

[Table of Contents](#)

Effects Icon. This is the icon shown at the bottom of the screen ingame when a spell is cast using this spell effect.

Description. This box shows the description for this spell effect.

Casting Effects. The **Sound** button brings up the **Sound** window. D-click a listed sound to associate it with the spell effect. The **Visual** drop down menu shows the visual effect associated with a spell effect. This file can be found under the Statics Tab in the Object window (the ID starts 'VFX_').

Bolt Effects. The **Sound** button brings up the **Sound** window. The **Visual** drop down menu shows the visual effect associated. The file for this can be found under the Weapons tab in the Object Window (the ID starts 'VFX_').

Hit Effects. The **Sound** button brings up the **Sound** window. The **Visual** drop down menu shows the visual effect associated. The file for this can be found under the Weapons tab in the Object Window (the ID starts 'VFX_').

Area Effects. The **Sound** button brings up the **Sound** window. The **Visual** drop down menu shows the visual effect associated. The file for this can be found under the Weapons tab in the Object Window (the ID starts 'VFX_').

Scale. The **Speed X** box is how fast the spell is cast. The **Size X** is how large the visual effects are when cast. **Size Cap** is the max size of the visual effect when cast.

Lighting Effect. The color associated with the light given off during a spell's casting.

The Sound Window

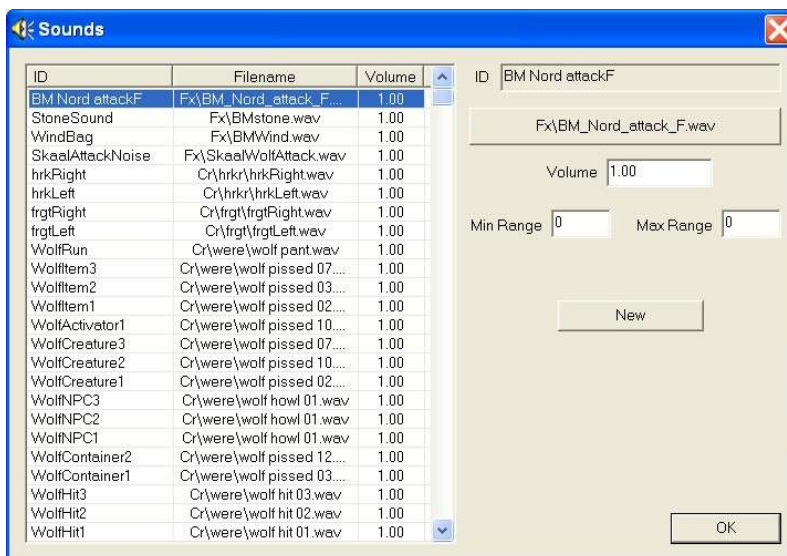
This window creates sounds using .wav. There may be issues with the properties of the audio file meeting certain criteria to be used in Morrowind. I have not identified if or what these values are yet. The large columned window has three columns in it:

ID. This must be unique.

Filename. The file used for the sound.

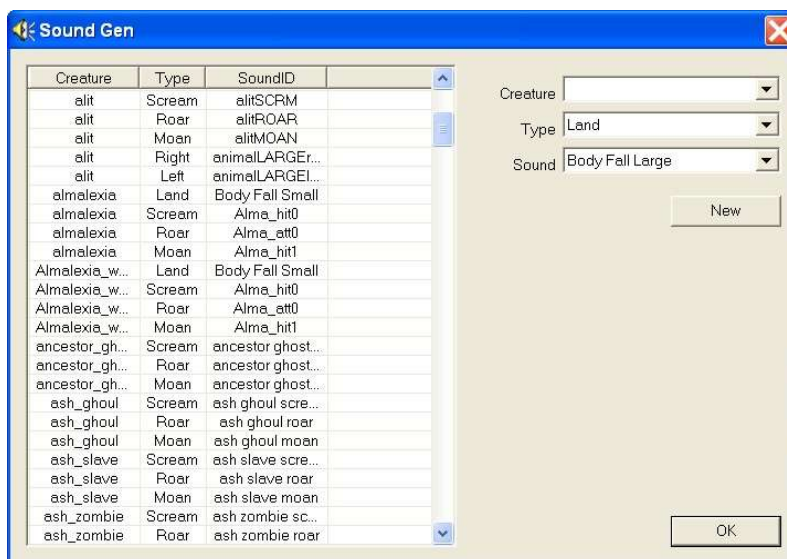
Volume. The volume at which the sound should be played.

To create a sound click New, enter the ID, click on the button to choose your .wav or .mp3 file to be used, and if you wish to, set the volume (it defaults to 1.00). Your new sound will appear in the window to the left. You can't change a sound's values once created. No big deal, just delete the sound instead (highlight in window and click delete), and create it again.



The Sound Generator Window

This window is what you use to assign sounds to creatures. Before you can do this though, you must have created the sound in the CS with the Sound Window (detailed above), and also have created the creature under the Creatures tab in the Object Window. This is easy. The big columned window has three columns in it. The first lists the creature assigned the sound. The second lists the action associated with that sound (i.e. the action of the creature when this sound should be played). The third column lists the ID of the sound. There are three drop down boxes to the right of this:



The third column lists the ID of the sound. There are three drop down boxes to the right of this:

Creature. Shows a list of all creatures under the creatures tab.

Type. Shows a list of all actions creatures can make that can be associated with a sound. Not every creature will make all sounds, choose just those that apply. Following is a description of sounds that can be assigned. You cannot create a new 'type' of sound.

Land. The sound a creature makes when it hits the ground. Choose either small, medium, or large.

Scream. The scream of a creature.

Roar. The roar of a creature.

Moan. The moan of a creature.

Swim Left. The splash sound of the left limb when swimming.

Swim Right. The splash sound of the right limb when swimming.

Left. The sound of the left 'foot' during walking or running.

Right. The sound of the right 'foot' during walking or running.

Left Wing. The sound of the left wing flapping during flight.

Right Wing. The sound of the right wing flapping during flight.

Sound. This shows a list of all available sounds to choose to assign to a creature's action.

Unlike other windows in the CS, you must define the sound assignment with the drop down boxes first, and then click on new for a new sound association. Click the drop down menus to pick a creature, the type of action, and the sound to assign. *Then* click on the new button. The new sound assignment will then appear in the window to the left. If you wish to change the sound already assigned to a creature action, delete what you wish to change first, then add what you want as a new sound. You do not have to assign a sound to a creature's actions, the creature will just not have a sound assigned and will be silent during such actions.

Chapter 5 Saving and Testing Your Mod

You finished your mod. Your plug-in is ready to be uploaded for others to enjoy. Did you test it? Did you clean it? What do I mean??... You got a little more work to do before you upload your mod. This chapter I will help you figure out if your mod is dirty and hopefully get as much testing done as possible, without causing problems for your mod.

Creating a Mod

It's so simple. The CS is so easy to use, right? Well, it actually is, despite how it may seem at first (ever worked with a professional auto-cad program?). When the CS first starts you have a blank world, almost no objects, an endless ocean, no races, factions, etc. The pull to create a new world is tempting for some (a Total Conversion). For others it is the chance to tweak the game to make it easier, or more difficult. To others it is the opportunity to expand an area of the game, basically, to make it more fun. The best thing you can do when starting is to take your idea, or vision, and write it down. Give it some focus. If your vision is just to create a new race, or make a house, that's pretty simple and you probably don't need to write out a mission statement. But if you're going to create a mod with new areas, NPCs, quests, items, etc. Write down the storyline, this will help you stay on track as to what you're doing, and why. If the idea changes, fine, go with it. But this will also keep your idea pure. If while working on one mod you get the idea for another, write it out and work on it separately. Unless you've got some experience creating mods, try not to combine ideas. Keep them separate, and if one would work well in the other you can always combine one with another. Keeping them separate can also help to keep the debugging process simpler. Working on just one project at a time also keeps you on track for completing it. If you work on three different mods at once, none will be done till you've completed all three. These are just suggestions, everyone has their own style of doing things. Do what works for you.

What is your base .esm(s)? Will your mod be dependent on only Morrowind? Maybe that and Tribunal? Bloodmoon? What patches do you have installed? Make sure you decide and note this in the readme.txt. Save some grief for those who want your mod, download it, and discover they can't play it because you didn't mention that you used `morrowind.esm` and `bloodmoon.esm` as the dependent files for your plug-in, but they only have Tribunal so far. If you want to make a name for yourself don't make it a bad one.

Take notes, lots of notes. Use a notebook to keep track. Write down any objects you create by ID, the changes you made (if any), what cell you put it in, or even the grid. Created some interiors? Keep a running tab of all objects used for that interior, scripts you've attached to objects (and what objects), reason for that interior (it's purpose). Making a quest? Write out a timeline of events and things that have to happen (and those that could happen), especially if you have more than one event taking place during the quest.

Test your plug-in. Most people do this, mostly because they want to play the great mod they created. But you have to do more than that. You have to explore every part of the mod you create. If you place an NPC in your mod with dialog, click on every response to verify you get every response you gave them. If you write a script, get it to activate and test every possible condition you wrote into it. If you create objects, use them and test them under every circumstance you can

think of. This is how you find bugs, and how others will find them if you don't when they play your mod. This takes time too, which is why we hail beta testers.

User created objects. Plan on using objects created by someone else you found in a mod you liked? Did you get permission to do so from the plug-in's author? The information on how to contact an author should be included with any plug-in [readme.txt](#) file. If one isn't included, where ever you downloaded the plug-in from should have had some kind of statement or at least the name of who submitted the plug-in. You should always make every attempt to get permission to use anything someone else created. If you don't and that author takes exception you could be sued. Plus, that's just showing respect for the work someone else did. Most are willing to let you 'borrow' their work if you just ask. Are you including new objects personally created by someone? Do you know how to find and include the correct `.nif`, `.kif`, `.tga`, `.dds`, or `.bmp` files in the package to upload? This could be a problem. If you forget to include the files that didn't come with the game, your mod won't work right. This is a bad thing (I've mentioned this before).

Compress the plug-in. Always compress the plug-in and any associated files needed by it. This makes it smaller so it is easier to upload, and also download. What utility you wish to use is up to you. The help feature of the utility should tell you how to create a zipped file and how to add files to it (refer [here](#) for what utility to use).

Beta Test it. Send it to some friends or find someone to do a test of your plug-in at a web site's forum. There are always people willing to beta test a plug-in. They can let you know what bugs and problems they find that you missed.

If you just create a simple plug-in with a few changes or things added you probably don't need to go through all this to verify that it works correctly and did everything right. But it won't hurt if you do. These are things you can do to keep your plug-in clean and bug free, and the bigger it is, the more things you do in it, the greater the chance of problems. Why go through all the work you did only to have people upset because it's buggy, or unplayable. Avoid the bad karma, do the little extra time and effort to make sure your mod is good to go.

Doubling?! I usually call it &\$%^, stupid, #*&\$, ^&@...

You saved your plug-in. Loaded a save game to see how it's working now, and you found multiple copies of certain objects. What the hell? You just made a common mistake many modders make. When you save a game Morrowind stores information and objects in that game in the saved file. So when you play a plug-in, save it, make changes to it in the CS, and then load up that game and play some more, you may experience what many call 'doubling'. Though you could experience more than two copies of an object if you keep saving, changing, loading the same plug-in.

How to fix doubling. It's easiest to just delete your saved game and start over, but if that's too painful, do these steps in order:

1. Find an interior that you have made absolutely no changes to whatsoever. Once you've entered it, save your game. Then exit the game.
2. Start Morrowind and click on Data Files. Uncheck your plug-in so it is not loaded. Click Play.

3. From the 'choose game' menu (New, Load, etc) load your saved game that you want to fix. You will get messages saying there are errors offering the option to continue or quit. Just keep clicking on continue until they stop.
4. Once your game has loaded, **SAVE** it immediately. Exit the game. This saves your game without the changes from your mod. Your saved game is now ok to play.
5. You can now start a game up with your plug-in and test out any changes you've made to your heart's content, and when you're done ***do not save your game!*** This is what causes the problems. Groan all you want, but if you open your mod in the CS, make changes, then load the saved game, you most likely will run into this problem again.

Prevent Doubling. To prevent this from happening is easy. Start a game with your plug-in, save it at some convenient point, name it 'Test' or whatever, and you're good. Anytime you wish to test out part of your plug-in load up the 'Test' game and don't save it when you're done testing your plug-in. If you accidentally do (I've done this), just fix it as above or start a new game and save a fresh clean game.

What's with the little * thingy?

It's called an asterisk. What it does though is let you (or anyone) know what you have changed in your plug-in. If you see any object's ID name with an asterisk next to it, it means at some point you modified or at least did something to that object. If you didn't make a change, but the object has an asterisk, congratulations, you have a dirty plug-in. How to fix? Make a note of any such objects you know you don't want any changes to, and clean it out by using the **Details** button on the [Data Files](#) window, or use a utility if you wish. Read on below under the *Cleaning the Unclean* section.

Note: *Only the active plug-in will display an asterisk next to objects that have been altered, but all objects will be listed in the Objects Window for all plug-ins loaded, active or not.*

Cleaning the Unclean

Clean your mod. Sounds like something your mother might say. But it needs to be done. A dirty plug-in is usually larger than necessary (as in file size). This means it takes longer to download, and a big file may be a factor for some choosing what to download. Someone who isn't diligent enough to clean a plug-in most likely isn't diligent enough to debug either (that's my opinion though), and we all love buggy games, don't we (**cough, cough*... daggerfall... *cough**). There are several utilities that can be used to help clean a plug-in (see [Chapter 6](#)), or you can use the CS itself.

To Clean. Start the CS, *do not load your plug-in*. Open the Data Files window and find your plug-in, highlight it. Do not click on any other files. Click on the 'Details' button at the bottom of the window. This will bring up a new window. This window allows you to see all the changes you've made to your mod. On the far left is a column that states what the changed object is. The fourth column shows the object ID. The fifth column is the offset, if you don't like to hex edit files, don't worry about it. If you see an object you do not want in your mod, or an object you didn't mean to make a change too (a spec of dirt I guess you could say), highlight the object you wish to remove from your mod. Now tap your 'delete' key. The CS will prompt you, stating that this object's *change* will be *ignored* and *not loaded* if you continue and load the plug-in,

effectively deleting this change. For this to work you **MUST** load your plug-in after marking all objects you want purged, and then save it! This is important! This is how you *clean* your mod of mistakes you've made by making changes to things you had not meant to. Want to increase that negative karma hanging over your head? Don't clean your mods.

The first column is your clue as to what an object is. Some of these are obvious. Here is a list and what they refer to:

ACTI	Object is an activator.
ALCH	Object is an alchemy potion.
APPA	Object is an apparatus.
ARMO	Object is a piece of armor.
BODY	Object is a body part, could be for a race, clothing, or armor.
BOOK	Object is a book
BSGN	Object is a birthsign.
CELL	Object is a cell. This is for the cell and all things within it.
CLAS	Object is a class.
CLOT	Object is a piece of clothing.
CONT	Object is a container.
CREA	Object is a creature.
DIAL	Object is a dialog topic.
DOOR	Object is a door.
ENCH	Object is an enchantment.
FACT	Object is a faction
GMST	Object is a Game Setting
INFO	Object is for a dialog statement.
INGR	Object is an ingredient.
LAND	Change in a cell's landscape (all of the landscaping within a cell).
LEVC	Object is a leveled creature (list).
LEVI	Object is a leveled item (list).
LIGH	Object is a light.
LOCK	Object is a lockpick.
LTEX	Object is to a landscape texture.
MGEF	Object is a magic effect.
MISC	Object is a miscellaneous item.
NPC_	Object is an NPC.
PGRD	Object is to a path grid.
PROB	Object is a probe.
REFR	Object is a reference (of an object).
REPA	Object is a repair item.
SCPT	Object is a script.
SKIL	Object is a skill.
SNDG	Object is a sound generated.
SOUN	Object is a sound.
SPEL	Object is a spell.
STAT	Object is a static.
WEAP	Object is a weapon.
RACE	Object is a race.

I believe I have listed almost all, if not all.

The second column shows objects that are part of an `.esm` file that has been deleted in your plug-in. You can't actually delete an object from an `.esm` file, so the object is still present in the `.esm`, but will always be shown that you deleted it in your plug-in (deleting *objects* from an `.esm` is not a good idea, deleting *references* is better. Try to avoid deleting objects that are part of an `.esm`. Seriously). When your plug-in is loaded by the game it will not load this object. Objects created within a plug-in, not part of an `.esm`, are truly deleted and do not show up in this column.

The third column shows those that you wish to 'ignore' (delete). These are the objects and changes you wish to remove from an `.esp`.

***Note:** References cannot be deleted from here. You do that by loading an `.esp`, and then opening a cell where the reference is placed, select it, then click delete. Or you could just delete the parent object from the object window (if user created, not part of an `.esm`).*

Viewing a Save file. It's possible to view the contents of your saved game with the CS. Remember, as you tour about Vvardenfel, the game keeps track of everything you do, and everything you come into contact with, and everywhere you go. All this information is stored in a Save file. But let's say you want to see the contents of a save file. All you have to do is copy the saved file to the `Data Files` folder, change the file extension from `.ess` to `.esp`. Now open the CS, open the `Data Files` window and find your save. Highlight it and click the Details button. **DO NOT LOAD IT!** (loading it may corrupt it) In the details window you can now see data from the save. You can even highlight a line of data and hit the delete button, setting a flag for that data to be ignored the next time it is loaded for playing by the game. Now just close the CS without loading the file, open the `Data Files` folder back up, change the file from an `.esp` back to `.ess`, and move it to the Save folder. Now you can play it again. Be very careful doing this! If you don't know exactly what you are doing, don't do this! But there are times when this is a useful way to clean out problem entries from a saved game that cause problems. Again, I can only stress, don't do this just to do it, you might just ruin your game! This is most useful to fix a game that a mod has made changes to it that breaks a quest or ruins dialog options. This sometimes can fix that, but only if you know exactly what to fix. You've been warned.

Respecting things other people make

If you wish to use something done by someone else, it's probably ok. But show some respect. They worked hard to make it, and want some recognition for having done so. Don't just add something you found in another plug-in to yours off handedly. Know who made the plug-in, and make an effort to contact them and ask for permission to include their work in your own. Often you will find a `readme.txt` file that will state that the author of a plug-in will allow this without contacting them. Usually there are certain limitations, such as including giving them credit for what you are including in your plug-in. Some may want you to do more. This is important, for while many may be willing to overlook a mod that included an object they created, some may not. There are plug-ins floating around the net that contain objects created by professionals using expensive software. If you don't credit them for their work and you use it in your plug-in, they could take legal action against you. I admit, I have yet to hear of a single case of this. But it's also a matter of respect. You may wonder why I included this section. Because I've found plug-in

readme.txt files stating they got an object from this plug-in called 'something' and another object from some website they can't remember the name of. Or they used some scripts they liked from another mod without specifically naming the plug-in or it's author. Simply put, I think that's bullshit. Show some respect, and give credit where it's due.

Chapter 6 Utilities

In this chapter I will try to give some insight into some utilities I have found, what they can be used for, usefulness, and maybe a couple tips to using them. This will not be a tutorial on how to use any of these. This will be just some basic information. By including this chapter I am not in any way endorsing any of these utilities, merely providing some information about them. Make your own judgments about whether they would be of any use to you, and if you should download and use them.

Bethesda Softworks Archive Browser (BSABrowser) - by Argent

This can be used to view, unpack, or repack a .bsa file. Even pack a new .bsa file.

BookGen (Book Generator utility) - by Argent

Write the text of a book in HTML or text format, then use to create the book for ingame use.

DDS Converter - by Bluehair

Use this to convert textures you've created into .dds files.

Enchanted Editor - by Farren

This utility can do about anything the TESCS can do (and more!) except landscaping and object reference placement. Excellent for object editing.

ESP Cleaner - Horatio

Simply what it says. Used to clean a dirty mod.

FPS Optimizer - by Alexander Stasenko

This can increase the visual distance and appearance of your game. Not really for modding, but it's just so cool.

GenMods - by Jim Adam

Create interiors quickly, even fill them with objects.

GMST Cleaner - by Argent

Use this to get rid of the evil game setting changes that can ruin your plug-in.

Landscape Generator - by Gilles_K

You can use this utility to quickly generate your landscape for your plug-in.

Leveled List Merger - by Horatio

This will create a single master leveled list for all plug-ins chosen so they all can be used.

Morrowind Clothing Catalog - by Klinn

[Table of Contents](#)

[Table of Contents](#)

Displays clothing objects as they would be worn, instead of folded up as in the CS.

MWEdit (Morrowind Editor) - by Dave Humphries

An alternate plug-in editor. You can do anything with it that the TESCS can do, except work with landscape and placing object references in the game.

Morrowind Script Generator - by Kaos_nyrb

Simple utility that can generate stock scripts for you. You type in object and variable data, and it spits out a script you can use. Will not create anything complex or long.

NIFDuplicator - by Ghostwheel

This utility can duplicate a .nif file. Great to use to make a mesh for your own textures.

NIFTexture - by Dave Humphries

Use this to figure out what textures are used by a mesh so you can retexture an object. You can also use it to associate a texture file with a mesh.

Overunity - by Shanjaq

This is one hell of a useful utility. You can use this to see what files your plug-in uses and what .esp or .esm files they belong to. Even package your plug-in. Very useful.

The Complete Guide to Tamriel Lore - by Mortis

Not a utility, but a compilation of reading material gathered from many various sources. A must have to understand the background behind much of the TES lore.

The Elder Scrolls Files (TESFiles) - by Mental Elf

Use this utility to pack your mod. It will identify all the necessary files to play your plug-in that are not contained in the Morrowind, Tribunal, or Bloodmoon. Then put them in a folder together, so you can then just zip them up together with the .esp and your readme.

The Elder Scrolls Construction Set (TESCS) Item Index - by Srikandi

This is a great resource. Use it to figure out what object you need in your mod, and under what tab it appears in the Object Window. Save yourself time searching for it.

The Elder Scrolls Advanced Mod Editor (TESAME) - by Erik Benerdal (aka Scarabus)

This can be used to clean a plug-in of dirty objects and merge mods together. You can also use it export objects from a plug-in, then import them into another.

The Elder Scrolls Dependency Tool Kit (TESDTK) - by Aaron White (aka Argent)

This can be used to make an .esp dependent on one or more master files, or to remove such dependency. It has other functions too.

The Elder Scrolls Faith utility (TESFaith) - by Paul Halliday (aka Lightwave)

Command line program enabling you to move, copy, or delete exterior cells singularly, or enmass, in one shot.

The Elder Scrolls Mod Utility (TESMU) - by Macgyver

This will install a plug-in, and any included files, and keep track of them. Also can delete a plug-in without removing those files shared with another plug-in.

The Elder Scrolls Plug-In Conflict Detector (TESPCD) - by Ely VanReen Soto

This will detect conflicts and problems between plug-ins.

The Elder Scrolls Tool (TESTool) - by Ghostwheel

This utility can automatically clean a mod, merge objects and dialog, and more.

Difficulty slider vs. tweak mods

There are many various plug-ins available on the net that are constructed solely for the purpose of making the game easier, or more difficult. They do this by adjusting just about every aspect of the game in some way to maintain balance, but also to make playing the game easier/harder. The idea behind doing this is to improve the experience of playing the game. Many plug-ins have been made that introduce one or more items to the game to make playing the game easier (some ridiculously), but they do not actually change the *settings* of the game to do this (I refer to these types of plug-ins as Uber Mods, because they contain super items so you just about can't lose in any situation). Most of the plug-ins that make a game more difficult do so by adjusting the Game Settings ([Gameplay->Settings](#)). This takes a lot of work and some serious thinking to keep the game balanced. Some make other changes too. Many people like these because: 1) Many people seem to think the game was too easy to complete straight out of the box. 2) They like a challenge. What most forget or don't know about is that the game has a difficulty slider that can be found under Options ingame (push 'Esc' and click 'Options'). What does it do? Not much really, it increases the hit points of creatures in the game so they are tougher. Whoopee. If you are one of those who like a challenge, I recommend considering downloading and installing one of the many tweak mods out there on the net. They can be found on most websites that host a download section.

Appendix A References and Tutorials

Here I will list some references I used to glean information from. Most I downloaded at some point in the past just to learn more about what I was trying to do within the CS. Some are repetitive, others are unique in the information they provide. A few didn't necessarily provide much new information for me, but may be of use to another. These are all tutorials or references put together by someone else and provided for download within the gaming community. I do not remember where I may have acquired many of them, so I am not including where to get them. Some contain this information already. There are many, many tutorials out there if one just looks for them. If one doesn't have the info you are looking for, keeping looking, you will find it eventually.

Morrowind Mod Maker's Bible v1.14

Compiled by Nether Void

Scripting For Dummies v8

by GhanBuriGhan

NPC Creation Tutorial

by Scott Fischer

TESCS Tutorial

by Scott Fischer

Various Tutorials

by Shadowsong

Various Tutorials

by Dragonsong

Dialog Tutorial

by Srikandi

Dialog Tutorial

by Calislahn

There are many many others to be found, but these are the ones that I found to be a good start, and helped me the most. The Morrowind official site forum (under Construction Set) also has many linked online tutorials.

Appendix B Tables

In this section I will show tables from within the CS and try to provide some explanation of what they are, and possibly how to use them, or the consequences of altering data within them.

Races List

There is a script included by Bethesda that can be used ingame to 'get' the PC's race. This script processes at the start of a new game and stores the race as the global variable 'PCRace'. The numerical value of the variable states the race:

1 = Argonian 3 = Dark Elf 5 = Imperial 7 = Nord 9 = Redguard
2 = Breton 4 = High Elf 6 = Khajiit 8 = Orc 10 = Woodelf

Spell Effects List

This is simply a listing of all spell effects that are used to create spells or for enchanting in the CS. The number represents its ID for the script function [RemoveEffect](#).

85 = sEffectAbsorbAttribute	20 = sEffectDrainFatigue	77 = sEffectRestoreFatigue
88 = sEffectAbsorbFatigue	18 = sEffectDrainHealth	75 = sEffectRestoreHealth
86 = sEffectAbsorbHealth	21 = sEffectDrainSkill	78 = sEffectRestoreSkill
89 = sEffectAbsorbSkill	19 = sEffectDrainSpellpoints	76 = sEffectRestoreSpellPoints
87 = sEffectAbsorbSpellPoints	126 = sEffectExtraSpell	42 = sEffectSanctuary
63 = sEffectAlmsiviIntervention	8 = sEffectFeather	3 = sEffectShield
47 = sEffectBlind	14 = sEffectFireDamage	15 = sEffectShockDamage
123 = sEffectBoundBattleAxe	4 = sEffectFireShield	46 = sEffectSilence
129 = sEffectBoundBoots	117 = sEffectFortifyAttackBonus	11 = sEffectSlowFall
127 = sEffectBoundCuirass	79 = sEffectFortifyAttribute	58 = sEffectSoultrap
120 = sEffectBoundDagger	82 = sEffectFortifyFatigue	48 = sEffectSound
131 = sEffectBoundGloves	80 = sEffectFortifyHealth	67 = sEffectSpellAbsorption
125 = sEffectBoundLongbow	83 = sEffectFortifySkill	106 = sEffectSummonAncestralGhost
121 = sEffectBoundLongsword	81 = sEffectFortifySpellpoints	110 = sEffectSummonBonelord
122 = sEffectBoundMace	52 = sEffectFrenzyCreature	134 = sEffectSummonCenturionSphere
130 = sEffectBoundShield	51 = sEffectFrenzyHumanoid	103 = sEffectSummonClannfear
124 = sEffectBoundSpear	16 = sEffectFrostDamage	104 = sEffectSummonDaedroth
7 = sEffectBurden	6 = sEffectFrostShield	105 = sEffectSummonDremora
50 = sEffectCalmCreature	39 = sEffectInvisibility	114 = sEffectSummonFlameAtronach
49 = sEffectCalmHumanoid	9 = sEffectJump	115 = sEffectSummonFrostAtronach
44 = sEffectCharm	41 = sEffectLight	109 = sEffectSummonGreaterBonewalker
132 = sEffectCorpus	60 = sEffectMark	102 = sEffectSummonScamp
70 = sEffectCureBlightDisease	43 = sEffectNightEye	107 = sEffectSummonSkeletalMinion
69 = sEffectCureCommonDisease	13 = sEffectOpen	116 = sEffectSummonStormAtronach
73 = sEffectCureParalyzation	27 = sEffectPoison	135 = sEffectSunDamage
72 = sEffectCurePoison	56 = sEffectRallyCreature	1 = sEffectSwiftSwim
22 = sEffectDamageAttribute	55 = sEffectRallyHumanoid	59 = sEffectTelekinesis
23 = sEffectDamageHealth	68 = sEffectReflect	133 = sEffectVampirism
24 = sEffectDamageMagicka	100 = sEffectRemoveCurse	0 = sEffectWaterBreathing
26 = sEffectDamageSkill	95 = sEffectResistBlightDisease	2 = sEffectWaterWalking
57 = sEffectDispel	97 = sEffectResistPoison	35 = sEffectWeaknessToPoison
62 = sEffectDivineIntervention	92 = sEffectResistShock	30 = sEffectWeaknessToShock

[Table of Contents](#)

[Table of Contents](#)

Scripting Command/Function List

Here I present a table showing all scripting commands (that I know), to 'use', and any errata associated with it. This is not an explanation of scripting. This is merely for reference, refer to [Scripting](#) for more detailed explanations of any command or function.

Command/Function	Use:	Errata
Activate	Activate, "object_ID"	Works on actors, containers, and activators, but not doors.
AddItem	AddItem "object_ID" #	The # is the number of items you want added. 65,535 is max
AddSoulGem	AddSoulGem "creature_ID", "soulgem_ID"	Will add a soulgem filled with named creature
AddSpell	AddSpell, "spell_ID", "target_ID"	Will add 'spell' to the PC's or NPC's spell inventory, but if 'spell' is curse or disease, it will affect them.
AddToLevCreature	AddToLevCreature "leveled_creature_list_ID" "creature_ID" "level"	Will add a creature to a leveled list for the specified "level"
AddToLevItem	AddToLevItem "leveled_item_list_ID" "item_ID" "level"	Will add an item to a leveled list for the specified "level"
AddTopic	AddTopic, "topic"	Will add a topic for dialog
AIActivate	AIActivate, "object_ID"	Actor will 'activate' an object
AIEscort	AIEscort, "actor_ID", "duration", "x-grid", "y-grid", "z-grid"	Actor will escort "actor_ID" to specified x, y, z grid
AIEscortCell	AIEscortCell, "actor_ID", "cell_ID", "duration", "x-grid", "y-grid", "z-grid"	Actor will escort "actor_ID" to specified "cell_ID" to x, y, z grid
AiFollow	AiFollow, "actor_ID", "duration", "x-grid", "y-grid", "z-grid"	Works the same as AIEscort
AiFollowCell	AiFollowCell, "actor_ID", "cell_ID", "duration", "x-grid", "y-grid", "z-grid"	Works the same as AIEscortCell
AiTravel	AiTravel, "x-grid", "y-grid", "z-grid"	Actor travels to x, y, z grid
AiWander	AiWander, "range", "duration", "time", "idle2", "idle3", "idle4", "idle5", "idle6", "idle7", "idle8", "idle9"	Actor will wander about the Path Grid, if no grid then randomly
BecomeWerewolf	Player->BecomeWerewolf	Turns actors/PC into werewolf
Begin	Begin "scriptname"	Start point of any script
Cast	Cast, "spell_ID", "target_ID"	Casts spell upon target
CellChanged	CellChanged = #	The # is a 1 or 0 1 = PC changed cells 0 = PC still in same cell
CellUpdate	BROKEN	Do not use.
CenterOnCell	CenterOnCell, "cell_ID"	Places actor in center of named cell
CenterOnExterior	CenterOnExterior, "x-grid", "y-grid"	Places actor in exterior at grid "x", "y"

ChangeWeather	ChangeWeather, "region_ID", #	The # is a value for the weather type: 0 = Clear 1 = Cloudy 2 = Foggy 3 = Overcast 4 = Rain 5 = Thunder 6 = Ash 7 = Blight
Choice	Choice, "button1", "button2"	Used in dialog, up to 5 button 'choices' can be designated
Clear- -ForceJump -ForceMoveJump -ForceRun -ForceSneak	actor->ClearForceJump, etc.	These will prevent the actor from executing the prior called command.
ClearInfoActor	ClearInfoActor	Used in dialog to prevent a journal entry (why?)
companion	companion = #	Must state short variable; allows PC to share equipment 1 = share 0 = not share
Console	N/A	Push the ~ (tilde) button to the left of the your 1 key (top left 1) to bring up the console, many commands can be run from the console (but not all)
CreateMaps	CreateMaps, "plug-in.esp"	Must first create a 'Maps' file in MW folder, and also enable command in morrowind.ini file. 0 = None 1 = Xbox 2 = Exterior Cell Maps
Day	Day	Global short variable, returns day of the month
DaysPassed	DaysPassed	Global short variable, returns number of days since game began
Disable/Enable	"object_ID"->Disable "object_ID"->Enable	Disable will tell the game to stop rendering an object (vanishes), Enable will tell the game to render a disabled object. Object can be an item or actor. Scripts still process when object is disabled
Disable/EnableLevitation	DisableLevitation EnableLevitation	If disabled, the spell effect levitation will not work
Disable/EnablePlayerControls	DisablePlayerControls EnablePlayerControls	If disabled, player cannot move, change view, or rotate viewpoint
Disable/EnablePlayerFighting	DisablePlayerFighting EnablePlayerFighting	If disabled, player cannot enter combat mode
Disable/EnablePlayerJumping	DisablePlayerJumping EnablePlayerJumping	If disabled, player cannot Jump

Disable/EnablePlayerLooking	DisablePlayerLooking EnablePlayerLooking	If disabled, player cannot rotate view with mouse
Disable/EnablePlayerMagic	DisablePlayerMagic EnablePlayerMagic	If disabled, player cannot cast spells
Disable/EnablePlayerViewSwitch	BROKEN	Use Disable/EnableVanityMode instead
Disable/EnableTeleporting	DisableTeleporting EnableTeleporting	If disabled, player cannot teleport
Disable/EnableVanityMode	DisableVanityMode EnableVanityMode	If disabled, player cannot change to 3 rd person viewpoint
DontSaveObject	DontSaveObject	Object will not be saved to player's saved game
Drop	Drop, "object_ID", #	Actor will drop object at their feet, # is number dropped
DropSoulgem	DropSoulgem, "creature_ID"	Object will drop a filled soulgem with 'creature' in it
Enable- -BirthMenu -ClassMenu -LevelUpMenu -NameMenu -Rest -RaceMenu -StatReviewMenu	EnableBirthmenu, etc.	Will open the appropriate menu for input by player. Do not open the Race menu more than once per period of game play. Save, exit game, reload, then open again.
Enable- -InventoryMenu -MagicMenu -MapMenu -StatsMenu	EnableInventoryMenu, etc.	If menu is disabled, will allow player to view menu again.
Equip	Equip, "object_ID"	Actor will equip the object
ExplodeSpell	ExplodeSpell, "spell_ID"	Reference will cast spell at itself
Face	Face, "x-grid", "y-grid"	PC will turn and face toward the given x, y grid, then continue (whatever it was doing)
FadeIn/Out	FadeIn, # FadeOut, #	The # is seconds, from 0-10, for screen to fade to black, or from black
FadeTo	FadeTo, %, #	The % is amount of darkness (black) screen fades to, from 0-100, the # is seconds spent fading
Fall	Fall	Object will fall to ground, or object below
FillJournal	FillJournal	PC's journal will fill with all entries, this may take some time
FillMap	FillMap	Shows all towns on world map
FixMe	Fixme	PC is moved 128 units in 'facing' direction, good to get unstuck
ForceGreeting	ForceGreeting	NPC will initiate dialog, regardless of where PC is in world
ForceJump	ForceJump	Forces NPC to jump in place
ForceMoveJump	ForceMoveJump	Forces NPC to jump while moving

ForceRun	ForceRun	Forces NPC to run when they move
ForceSneak	ForceSneak	Forces NPC into sneak mode
GameHour	GameHour	Float variable returns the time of day (0-23)
Get- -ForceJump -ForceMoveJump -ForceRun -ForceSneak	GetForceJump, etc	Checks if actor is under named command 0 = false 1 = true
Get/Mod/Set -Alarm -ArmorBonus -AttackBonus -Attributes -Blindness -CastPenalty -Chameleon -DefendBonus -Disposition -Fatigue -Fight -Flee -Flying -Health -Hello -Magicka -Paralysis -PCCrimeLevel -PCVisionBonus -Reputation -ResistBlight -ResistCorprus -ResistDisease -ResistFire -ResistFrost -ResistMagicka -ResistNormalWeapons -ResistParalysis -ResistPoison -ResistShock -Silence -Skills -SuperJump -SwimSpeed -WaterBreathing -WaterWalking	GetAlarm, etc.	Returns a float value when called
	ModAlarm, # etc.	Changes the amount by #, cannot change beyond min/max
	SetAlarm, # etc.	Sets the amount to #, even if beyond normal min/max
Get/Mod/SetInvisible	GetInvisible, etc.	Returns value if invisible. Note this was spelled: GetInvisibile prior to Tribunal release 0 = false 1 = true

Get/Mod/SetFactionReaction	GetFactionReaction, "faction_ID1", "faction_ID2"	Returns reaction value of faction1 for faction2
	ModFactionReaction, "faction_ID1", "faction_ID2", #	Changes reaction value of faction1 for faction2 by amount of #
	SetFactionreaction, "faction_ID1", "faction_ID2", #	Sets reaction value of faction1 for faction2 to amount of #
Get/Mod/SetPCFacRep	GetPCFacRep, "faction_ID"	Returns value of PC's reaction bonus/penalty for faction
	ModPCFacRep, "faction_ID", #	Changes reaction value of PC for faction by amount of #
	SetPCFacRep, "faction_ID", #	Sets reaction value of PC for faction to amount of #
Get/Mod/SetScale	GetScale	Returns value of size
	ModScale, #	Changes size by amount of #
	SetScale, #	Set size to amount of #, this can change size beyond normal min/max allowed in CS
Get/Mod/SetWaterLevel	GetWaterLevel	Returns z-axis of water level
	ModWaterLevel, #	Changes z-axis of water level by amount of #
	SetWaterLevel, #	Set z-axis of water level to level of #
Get/SetJournalIndex	GetJournalIndex, "journal_ID" #	Returns last (maybe highest?) entry for named journal ID
	SetJournalIndex, "journal_ID" #	Sets a named journal entry to #, suggest using "Journal" instead
Get/SetLevel	GetLevel	Returns PC or actor level
	SetLevel	Sets level or actor or PC
Get/SetPos	GetPos, "x, y, or z axis"	Returns grid of named axis
	SetPos, "x, y, or z axis", #	Sets object to named axis at grid #
GetAIPackageDone	GetAIPackageDone	Returns a value of 1 for one frame when AiPackage finishes
GetAngle	GetAngle, "axis"	Returns a value in degrees (of 360) or the orientation of an object along named axis
GetArmorType	GetArmorType, "value_a" 'value_a' can be: helmet = 0 cuirass = 1 left pauldron = 2 right pauldron = 3 greaves = 4 boots = 5 left gauntlet = 6 right gauntlet = 7 shield = 8 left bracer = 9 right bracer = 10	Returns value for the 'weight class' of the called armor piece: -1 = unarmored 0 = light armor 1 = medium armor 2 = heavy armor
GetAttacked	GetAttacked	Returns value stating if PC has ever attacked actor 0 = false 1 = true

GetBlightDisease	GetBlightDisease	Returns value stating if PC has a blight disease 0 = false 1 = true
GetButtonPressed	GetButtonPressed	Returns value stating which button was pressed: -1 = None 0 = button1 1 = button2, etc.
GetCollidingActor	GetCollidingActor	Returns value stating if any actor or PC is colliding with object: 0 = false 1 = true
GetCollidingPC	GetCollidingPC	Returns value stating if any actor or PC is colliding with object: 0 = false 1 = true
GetCommonDisease	GetCommonDisease	Returns value stating if PC has a common disease 0 = false 1 = true
GetCurrentAIPackage	GetCurrentAIPackage	Returns value stating which current AIPackage is active of an actor: -1 = None 0 = Wander 1 = Travel 2 = Escort 3 = Follow 4 = Activate 5 = Pursue
GetCurrentWeather	GetCurrentWeather	Returns value stating current weather in cell: 0 = Clear 1 = Cloudy 2 = Foggy 3 = Overcast 4 = Rain 5 = Thunder 6 = Ash 7 = Blight
GetDeadCount	GetDeadCount, "actor_ID"	Returns value stating number of times actor has been killed
GetDetected	GetDetected, "actor_ID"	Returns value stating if actor can detect named 'actor' 0 = false 1 = true
GetDisabled	GetDisabled, "object_ID"	Returns value stating if object is disabled 0 = false 1 = true
GetDistance	GetDistance, "object_ID"	Returns distance of 'object' from calling object in units

GetEffect	GetEffect, "spell_effect"	Returns value stating if effect is active on calling actor 0 = false 1 = true
GetHealthGetRatio	GetHealthGetRatio	Returns a value stating percent level of Health: 0.0 = dead 0.4 = 40% of full 1.0 = 100% full, etc.
GetHealthRatio	BROKEN	Use GetHealthGetRatio instead
GetInterior	GetInterior	Returns value stating if PC is in an interior cell 0 = false 1 = true
GetItemCount	GetItemCount, "object_ID"	Returns value stating number of items in inventory
GetLineOfSight	GetLineOfSight, "actor_ID"	Returns value stating if actor is within line of sight (could be seen) of actor or PC. Same as GetLOS. 0 = false 1 = true
GetLocked	GetLocked	Returns value stating if object is locked or not 0 = false 1 = true
GetLOS	GetLOS	Returns value stating if actor is within line of sight (could be seen) of actor or PC. Same as GetLineOfSight 0 = false 1 = true
GetMasserPhase	GetMasserPhase	Returns value stating phase of moon. Interiors always return value of 0 0 = new (or interior) 1 = waxing or waning crescent 2 = waxing or waning half 3 = waxing or waning gibbous 4 = full moon
GetPCCell	GetPCCell, "cell_ID"	Returns value stating if actor or PC is in cell named 0 = false 1 = true
GetPCInJail	GetPCInJail	Returns value stating if PC is in jail 0 = false 1 = true
GetPCJumping	GetPCJumping	Returns value stating if PC is jumping 0 = false 1 = true

GetPCRank	GetPCRank	Returns value stating PC's rank in calling actor's faction 0 = false 1 = true
	GetPCRank, "faction_ID"	Returns value stating PC's rank in named faction 0 = false 1 = true
GetPCRunning	GetPCRunning	Returns value stating if PC is running 0 = false 1 = true
GetPCSleep	GetPCSleep	Returns value stating if PC is sleeping 0 = false 1 = true
GetPCSneaking	GetPCSneaking	Returns value stating if PC is sneaking 0 = false 1 = true
GetPCTravelling	GetPCTravelling	Returns value stating if PC is fast traveling (by silt strider or boat) 0 = false 1 = true
GetPlayerControlsDisabled	GetPlayerControlsDisabled	Returns value stating if Player Controls are disabled 0 = false 1 = true
GetPlayerFightingDisabled	GetPlayerFightingDisabled	Returns value stating if Player Fighting is disabled 0 = false 1 = true
GetPlayerJumpingDisabled	GetPlayerJumpingDisabled	Returns value stating if Player Jumping is disabled 0 = false 1 = true
GetPlayerLookingDisabled	GetPlayerLookingDisabled	Returns value stating if Player Looking is disabled 0 = false 1 = true
GetPlayerMagicDisabled	GetPlayerMagicDisabled	Returns value stating if Player Magic is disabled 0 = false 1 = true
GetPlayerViewSwitchDisabled	BROKEN	Use GetVanityModeDisabled
GetRace	GetRace, "race_ID"	Returns value stating if of stated race Returns value stating if Player Controls are disabled 0 = false 1 = true
GetSecondsPassed	GetSecondsPassed	Returns value of time passed during this frame (as a fraction of a second)

GetSecundaPhase	GetSecundaPhase	Returns value stating phase of moon. Interiors always return value of 0 0 = new (or interior) 1 = waxing or waning crescent 2 = waxing or waning half 3 = waxing or waning gibbous 4 = full moon
GetSecundusPhase	BROKEN	Use GetSecundaPhase
GetSoundPlaying	GetSoundPlaying, "sound_ID"	Returns a value if named sound is playing; 0 = false 1 = true
GetSpell	GetSpell, "spell_ID"	Returns value stating if 'spell' is in the actor's inventory (if a spell). If a curse or disease, returns value if actor is affected by it. 0 = false 1 = true
GetSpellEffects	GetSpellEffects, "spell_ID"	Returns value stating if actor or PC is affected by named spell 0 = false 1 = true
GetSpellReadied	GetSpellReadied	Returns value stating if actor or PC has a spell readied 0 = false 1 = true
GetSquareRoot	GetSquareRoot, #	Computes the square root of #
GetStandingActor	GetStandingActor	Returns value if actor or PC is standing on calling object 0 = false 1 = true
GetStandingPC	GetStandingPC	Returns value stating if PC is standing on calling object 0 = false 1 = true
GetStartingAngle	GetStartingAngle, "axis"	Returns value stating original orientation in degrees (360) actor started at on named axis
GetStartingPOS	GetStartingPOS	Returns value stating original grid actor started at on named axis
GetTarget	GetTarget, "actor_ID"	Returns value stating if target is named actor 0 = false 1 = true
GetVanityModeDisabled	GetVanityModeDisabled	Returns value stating if Vanity Mode is disabled 0 = false 1 = true

GetWeaponDrawn	GetWeaponDrawn	Returns value stating if a weapon is drawn and ready to attack 0 = false 1 = true
GetWeaponType	GetWeaponType	Returns value stating weapon type of actor's current readied weapon -1 = Unarmed 0 = short blade, 1-hand 1 = long blade, 1-hand 2 = long blade, 2-hand 3 = blunt, 1-hand 4 = blunt, 1-hand close 5 = blunt, 2-hand wide 6 = spear, 2-hand wide 7 = axe, 1-hand 8 = axe, 2-hand close 9 = bow 10 = crossbow 11 = thrown weapon 12 = arrow 13 = bolt
GetWerewolfKills	GetWerewolfKills	Returns value stating number of kills as werewolf (or last time was a werewolf), unsure of this one.
GetWindSpeed	GetWindSpeed	Returns value stating wind speed, always 0 indoors
Goodbye	Goodbye	Only for dialog, forces PC to end a dialog
GotoJail	GoToJail	PC is sent to prison
HasItemEquipped	HasItemEquipped, "object_ID"	Returns value stating if actor has named object equipped 0 = false 1 = true
HasSoulgem	HasSoulgem, "creature_ID"	Returns value stating if actor has soulgem with named creature in inventory 0 = false 1 = true
Help	Help	Only for the console. List some of the commands available only in the console
HitAttemptOnMe	HitAttemptOnMe, "object_ID"	Returns value stating if an attack was attempted with named object 0 = false 1 = true
HitOnMe	HitOnMe, "object_ID"	Returns value stating if hit by named object 0 = false 1 = true

HurtCollidingActor	HurtCollidingActor, #	Modifies the health of any actor colliding with object by # Negative # damages health Positive # heals health
HurtStandingActor	HurtStandingActor, #	Modifies the health of any actor standing on object by # Negative # damages health Positive # heals health
IsWerewolf	IsWerewolf	Returns value if PC is a werewolf 0 = false 1 = true
Journal	Journal, "ID", #	Adds an entry to the PC's journal, named 'ID', and given a number according to #
Lock	Lock #	Locks calling object at # level
LoopGroup	LoopGroup, "groupname", #	Plays 'groupname' animation, # number of times in a row
LowerRank	LowerRank	PC is lowered one rank in Faction
MenuMode	MenuMode	Returns value if PC is viewing any menu 0 = false 1 = true
MessageBox	MessageBox	Displays text at mid-bottom of screen. Can also display button choices for PC to answer a question (then also use GetButtonPressed). Can enter other info to be displayed in text: ^PCName = player's name ^PCClass = player's class ^PCRace = player's race ^PCRank = player's rank ^NextRank = next rank ^Cell = PC's current cell ^Global = any global variable ^NPC.name = NPC name ^NPC.race = NPC race ^NPC.class = NPC class ^NPC.faction = NPC faction ^NPC.rank = NPC rank
minimumprofit	minimumprofit	Variable for determining if starting value of goods and gold for a hireling is greater than current
ModCurrent- -Fatigue -Health -Magicka	ModCurrentFatigue, etc.	Changes value of stat, but not above max normal value
ModRegion	ModRegion, "region_ID", "clear", "cloudy", "foggy", "overcast", "rain", "thunder", "ash", "blight"	Change the percent chance of a weather type in a named region, replace weather type named with numerical percent you want

Month	Month	Global variable for the month of the year, 0-11. BUG: the game ends month 11 and then goes to month 1 (skips 0)
Move	Move, "axis", #	Moves an object along defined 'axis' at # units per second
MoveOneToOne	MoveOneToOne	I do not know this command's function
MoveWorld	Move, "axis", #	Works same as Move.
OnActivate	OnActivate	Returns value if object has been activated 0 = false 1 = true
OnDeath	OnDeath	Returns value for one frame if actor dies 0 = false 1 = true
OnKnockout	OnKnockout	Returns value for one frame if actor is knocked out 0 = false 1 = true
OnMurder	OnMurder	Returns value for one frame if actor is murdered. Will return a true value for both OnDeath and OnMurder if both in same script, use only one of them. 0 = false 1 = true
OnPCAdd	OnPCAdd	Global variable that returns value if object is added to inventory. Must be reset to 0 manually in script 0 = false 1 = true
OnPCDrop	OnPCDrop	Global variable that returns value if object is dropped from inventory. Must be reset to 0 manually in script 0 = false 1 = true
OnPCEquip	OnPCEquip	Global variable that returns value if object is equipped and remains equipped. Must be reset to 0 manually in script 0 = false 1 = true
OnPCHitMe	OnPCHitMe	Global variable that returns value if PC hits actor. Must be reset to 0 manually in script 0 = false 1 = true

OnPCRepair	OnPCRepair	Global variable that returns value if item is repaired by PC. Must be reset to 0 manually in script 0 = false 1 = true
OnPCSoulGemUse	OnPCSoulGemUse	Global variable that returns value if PC uses a soulgem. Must be reset to 0 manually in script 0 = false 1 = true
OnRepair	BROKEN	Does not work.
PayFine	PayFine	Resets fines to crime level of PC to zero, also removes all stolen items
PayFineThief	PayFineThief	Resets fines and crime level of PC to zero, does not remove stolen items
PCClearExpelled	PCClearExpelled, "faction_ID"	PC is no longer expelled from named faction. If no faction stated, the actors
PCExpell	PCExpell, "faction_ID"	Expels the PC from named faction
PCExpelled	PCExpelled, "faction_ID"	Returns value if PC has been expelled from named faction 0 = false 1 = true
PCForce1stPerson	PCForce1stPerson	Forces the PC view to 1 st person when current PC animation finishes
PCForce3rdPerson	PCForce3rdPerson	Forces the PC view to 3 rd person when current PC animation finishes
PCGet3rdPerson	PCGet3rdPerson	Returns value if PC is in 3 rd person view 0 = false 1 = true
PCJoinFaction	PCJoinFaction, "faction_ID"	Joins the PC to the named faction, or the actor's faction
PCLowerRank	PCLowerRank, "faction_ID"	Lowers the PC's rank in the named faction
PCRace	PCRace	Global variable that states the PC's race
PCRaiseRank	PCRaiseRank, "faction_ID"	Raises the PC's rank in the named faction
PCSkipEquip	PCSkipEquip	Variable when placed on an item to prevent item from being equipped
PCVampire	PCVampire	Returns value if PC is, or has been, a vampire -1 = was a vampire (cured) 0 = not a vampire 1 = a vampire

PCWerewolf	PCWerewolf	Returns value if PC is, or has been, a werewolf (cured) -1 = was a werewolf 0 = not a werewolf 1 = a werewolf
PlaceAtMe	PlaceAtMe, "object_ID", #, "distance", "direction"	Places # number of named objects, 'distance' in units away from calling object/actor. 'Direction' is a value listed below 0 = front 1 = back 2 = left 3 = right
PlaceAtPC	PlaceAtPC, "object_ID", #, "distance", "direction"	Places # number of named objects, 'distance' in units away from PC. 'Direction' is a value listed below 0 = front 1 = back 2 = left 3 = right
PlaceItem	PlaceItem, "object_ID", "x", "y", "z", "facing"	Places named object at x, y, z grid. 'facing' is direction object faces in degrees (360). This only works in exterior cells
PlaceItemCell	PlaceItemCell, "object_ID", "cell_ID", "x", "y", "z", "facing"	Places named object at x, y, z grid, in named cell. 'facing' is direction object faces in degrees (360). This is only for interior cells. BUG: seems to be an issue with objects placed disappearing from a save game when loaded.
PlayBink	PlayBink, "movie_filename", #	Plays named .bik movie. # is a variable that can enable them to 'esc' from movie 0 = cannot escape 1 = can escape
PlayGroup	PlayGroup, "animation"	Plays named 'animation'
PlayLoopSound3D	PlayLoopSound3D, "sound_ID"	Plays named sound as if coming from calling object (will be quieter as PC moves away) continuously until stopped with the StopSound command
PlayLoopSound3DVP	PlayLoopSound3DVP, "sound_ID", "volume", "pitch"	Plays named sound as if coming from calling object (will be quieter as PC moves away) continuously until stopped with the StopSound command at assigned values of 'volume' and 'pitch'
PlaySound	PlaySound, "sound_ID"	Plays named sound as if coming from PC's location at normal volume

PlaySound3D	PlaySound3D, "sound_ID"	Plays named sound as if coming from calling object (will be quieter as PC moves away)
PlaySound3DVP	PlaySound3DVP, "sound_ID", "volume", "pitch"	Plays named sound as if coming from calling object (as above) at assigned values of 'volume' and 'pitch'.
PlaySoundVP	PlaySoundVP, "sound_ID", "volume", "pitch"	Plays named sound at PC's location at assigned values of 'volume' and 'pitch'.
Position	Position, "x", "y", "z", "facing"	Places object/actor/PC at x, y, z grid 'facing' the direction object faces in degrees (360), for use with exterior cells. Some issues with this function
PositionCell	PositionCell, "x", "y", "z", "cell_ID", "facing"	Places object/actor/PC at x, y, z grid, in named cell, facing the direction stated in degrees (360). For use with interior cell, but works for exterior also in you name an exterior cell
RaiseRank	RaiseRank	Raises actor in faction rank
Random	Random, #	Game generates a random number from 0 to # (i.e. if # = 99, the game generates a number between 1-100)
RemoveEffects	Remove Effect, "effect_ID"	Removes spell effect from 'object', must use numerical ID assigned to an effect, not name
RemoveFromLevCreature	RemoveFromLevCreature "leveled_creature_list_ID" "creature_ID" "level"	Will remove a creature from a leveled list for the specified "level"
RemoveFromLevItem	RemoveFromLevItem "leveled_item_list_ID" "item_ID" "level"	Will remove an item from a leveled list for the specified "level"
RemoveItem	RemoveItem, "object_ID" #	Removes # number of named objects from actor/PC inventory. There are issues with using this command
RemoveSoulgem	RemoveSoulgem, "creature_ID"	Removes one soulgem with named creature within it
RemoveSpell	RemoveSpell, "spell_ID"	Removes named spell from actor or PC's inventory. If 'spell' is a curse/disease will remove affliction
RemoveSpellEffects	RemoveSpellEffects, "spell_ID"	Removes effects of named spell affecting actor or PC
RepairedOnMe	RepairedOnMe, "repair_object_ID"	Returns value if calling object is repaired by named repair object 0 = false 1 = true
ResetActors	ResetActors	Returns all actors to starting positions and resets all AI packages to their default

Resurrect	Resurrect	Resets an actor to beginning default state. No changes made since game started will be present
Return	Return	Tells game to stop processing a script that frame
Rotate	Rotate, "axis" #	Rotates object by its own named axis a number of degrees/second equal to #
RotateWorld	RotateWorld, "axis" #	Rotates object by the named world axis a number of degrees/second equal to #
SameFaction	SameFaction	Returns value if PC is same faction as actor
Say	Say, "filename", "text"	Plays 'filename' and displays 'text'. The text usually displays what is being spoken by the 'filename'. Only works on currently animating objects
SayDone	SayDone	Returns value if object is done speaking, or 'filename' played with Say command is finished 0 = false 1 = true
ScriptRunning	ScriptRunning, "script_ID"	Returns value if script is currently processing 0 = false 1 = true
Set	Set "variable" to "(name_ID)"	Assign a value to a stated variable
SetAngle	SetAngle, "axis", #	Sets object to the degree angle stated by #, on the world stated axis
SetAtStart	SetAtStart	Resets object to original default start position
SetDelete	SetDelete, #	Deletes an object. Disable an object for 10 frames or so before you delete it. # is a flag 1 = delete object 0 = clear delete
SetWerewolfAcrobatics	SetWerewolfAcrobatics, #	Sets the Acrobatics skill level to # when in werewolf form
ShowGroup	ShowGroup	Unknown
ShowMap	ShowMap, "cell_ID"	Shows the map for the named cell, to include all cells that start with 'cell' name
ShowRestMenu	ShowRestMenu	Opens rest menu allowing PC to rest
ShowTargets	ShowTargets	Will show target group member in the console of selected actor

ShowVars	ShowVars	If entered in console, will list all global and local variables and their value. If an object is selected (centered), only those variables of that object will be listed
SkipAnim	SkipAnim	Causes the current animation to not play
Start/StopCombat	StartCombat, "actor_ID" StopCombat, "actor_ID"	Named actor will begin combat, or stop combat accordingly. Only call this once per script
Start/StopScript	StartScript, "script_ID" StopScript, "script_ID"	Will start or stop a script. Cannot use either command from the script you wish to start or stop
StayOutside	StayOutside	When used with a 'companion' tell them to remain in an exterior cell when the PC enters an interior. They will rejoin the PC upon return
StopCellTest	StopCellTest	Stops test started with TestCells
StopSound	StopSound, "sound_ID"	Ends the playing of named sound
StreamMusic	StreamMusic, "music_file"	Plays named music file
TestCells	TestCells	Tests all cells in game for problems, takes a while
TestInteriorCells	TestInteriorCells	Tests interior cells for problems
TestModels	TestModels	Tests 3D models in game for problems
ToggleAI	ToggleAI	Turns on/off processing of AI functions
ToggleBorders	ToggleBorders	Turns on/off exposure of cell borders
ToggleCollision	ToggleCollision	Turns on/off, when off objects cannot touch or interact, even PC
ToggleCollisionBoxes	ToggleCollisionBoxes	Turns on/off 'selection' boxes showing point of collision for objects
ToggleCollisionGrid	ToggleCollisionGrid	Displays the grid used to determine collision between objects
ToggleCombatStats	ToggleCombatStats	Turns on/off the display of combat stats in the console
ToggleDebugText	ToggleDebugText	Turns on/off the display of game stats
ToggleDialogStats	ToggleDialogStats	Turns on/off display of dialog values in the console as they process
ToggleFogOfWar	ToggleFogOfWar	Turns on/off hiding unknown areas on the game map
ToggleFullHelp	ToggleFullHelp	Displays all usable commands and their short cuts in the console

ToggleGodMode	ToggleGodMode	Turns on/off invulnerability of PC
ToggleGrid	ToggleGrid	Creates a mass of numbers on my screen and slows it to a crawl. Just don't toggle this one
ToggleKillStats	ToggleKillStats	Turns of/off tracking of the number of each creature type killed by the PC
ToggleLights	ToggleLights	Turns on/off light emitted by light objects, does not affect ambient light (not check yet with interiors)
ToggleLoadFade	ToggleLoadFade	Turns on/off fade effect in game
ToggleMagicStats	ToggleMagicStats	Turns on/off the display of spell processing in the console
ToggleMenus	ToggleMenus	Turns on/off all menus
TogglePathGrid	TogglePathGrid	Turns on/off display of the Path Grid
ToggleScripts	ToggleScripts	Turns on/off the processing of scripts
ToggleSky	ToggleSky	Turns on/off the display of the sky
ToggleStats	ToggleStats	Turns on/off the display of all stats in the console
ToggleVanityMode	ToggleVanityMode	Turns on/off the ability of the PC to switch to vanity mode
ToggleWater	ToggleWater	Does nothing in exterior cell, unsure about interior cells
ToggleWireFrame	ToggleWireFrame	Turns on/off the display of the world and objects as wireframe
ToggleWorld	ToggleWorld	Turns on/off the display of the world
TurnMoonRed	TurnMoonRed	Turns the moon red (affects werewolf afflicted PCs)
TurnMoonWhite	TurnMoonWhite	Turns the moon white (affects werewolf afflicted PCs)
UndoWerewolf	UndoWerewolf	Forces change from werewolf to normal for PC
Unlock	Unlock	Unlocks the calling object
UsedOnMe	BROKEN	Does not work
WakeUpPC	WakeUpPC	Wakes PC up if asleep
Year	Year	Global variable stating current year

Game Settings

These numbers and settings are what define every action within the game. Changing these will alter how the game functions. Don't mess with this unless you know exactly what you are attempting to do. If you do, I suggest only altering one setting, a little at a time till you achieve the effect you wish. The following paragraph and table was copied directly from *Scripting for Dummies v8* by GhanBuriGhan. All credit for this part goes to him (and if you find a mistake, blame me). I've done this so you don't necessarily have to open two references at once.

[Table of Contents](#)

[Table of Contents](#)

Paragraph prior to Game Settings table by GhanBuriGhan-

"The following long table is a list of game settings. Listed here are all settings that have a numerical value. Not listed are string entries – which are used to set many standard message texts, menu-texts, spell effect names, etc. But since they are fairly descriptive, they should be easy to figure out. Not so the numerical settings. Thanks to four forum members (maxpublic, Ldones, Wakim and Iudas), the meaning of many of the settings is now known and compiled into the list below. The list may still be a bit rough. I have not edited it thoroughly, but I am sure it will be interesting information for many modders.

In many cases where you see Base and Mult game settings, the formula they're involved in is a standard linear equation in the form $y = mx + b$, where m is the mult, b is the base, and x is some attribute, skill, or other value.

Note that the entry names begin with f or i (for floats and integers). The string entries begin with s (string). To my knowledge they can not be changed in-game, only in the TES-CS."

Nr.	Name	Value	Description and Comments
"465"	"fRepairMult"	1.0000	Sets the general effectiveness of the repair skill of the character, via the armorer's hammer used
"466"	"fRepairAmountMult"	3.0000	Tells the game how many points of health are returned to the item when repaired Determines cost for repairing items (Whether calculated from Max Item Health or Item Cost, I'm not sure)
"467"	"fSpellValueMult"	10.0000	
"468"	"fSpellMakingValueMult"	7.0000	
"469"	"fEnchantmentValueMult"	1000.0000	is the setting for the price you pay at an enchanter to enchant an item. Linear.
"470"	"fTravelMult"	4000.0000	Sets the cost of Silt Strider and boat travel (I think) Multiplies cost of Travel – Unsure why the number is so high, but raising it raises the cost of Fast Travel
"471"	"fTravelTimeMult"	16000.0000	Tells the game how much time elapses during this sort of travel
"472"	"fMagesGuildTravel"	10.0000	Sets the cost of Guild Guide travel
"947"	"fWortChanceValue"	15.0000	Iudas: Used to calculate whether a plant has any ingredients inside. Wakim: is compared to your alchemy skill to determine which of the effects of an ingredient you can see.
"949"	"fMinWalkSpeed"	100.0000	This is the minimum walking speed of the PC, regardless of stats, skills or encumbrance
"950"	"fMaxWalkSpeed"	200.0000	This is the maximum walking speed of the PC, regardless of stats, skills, or encumbrance The actual walking speed of NPC's (and the PC) is set by checking various fActors (Speed, Athletics, etc.) and assigning a value between fMinWalkSpeed and fMaxWalkSpeed based on that – The two settings dictate the spectrum of Walk Speeds
"951"	"fMinWalkSpeedCreature"	5.0000	The same as for the PC, but if you badly encumber a creature it'll move veeerrry slowly. I've done this by accident.
"952"	"fMaxWalkSpeedCreature"	300.0000	Same as above, they get faster, so they cover the speed spectrum more rapidly
"953"	"fEncumberedMoveEffect"	0.3000	This sets how encumbrance affects walking and running speed, within the min/max limits set by other values.
"954"	"fBaseRunMultiplier"	1.7500	Exactly as it says. Changing the value will increase/decrease base running speed. Dictates how much faster Running is than the current Walk Speed
"955"	"fAthleticsRunBonus"	1.0000	Sets how Athletics affects running speed.
"956"	"fJumpAcrobaticsBase"	128.0000	Sets the base jumping distance for the PC.
"957"	"fJumpAcroMultiplier"	4.0000	Sets the multiplier for Acrobatics, which is why you can leap over tall buildings when your Acro is high enough.
"958"	"fJumpEncumbranceBase"	0.5000	Effects how greatly jumping ability is effected by Encumbrance, but I'm unsure how
"959"	"fJumpEncumbranceMultiplier"	1.0000	Effects how greatly jumping ability is effected by Encumbrance, but I'm unsure how
"960"	"fJumpRunMultiplier"	1.0000	UNSURE – Presumably effects Jump Distance while running (it doesn't seem to effect height, but I could be wrong)
"961"	"fJumpMoveBase"	0.5000	
"962"	"fJumpMoveMult"	0.5000	
"963"	"fSwimWalkBase"	0.5000	Multiplies your walking speed to achieve the swimming speed at a 'walk' Base swim speed while 'walking'
"964"	"fSwimRunBase"	0.5000	Multiplies your running speed to achieve the swimming speed at a 'run'.

"965"	"fSwimWalkAthleticsMult"	0.0200	Tells the game how Athletics affects 'walking' swimming speed. These low values keep you from flying through the water like you do on land when your Athletics is high.
"966"	"fSwimRunAthleticsMult"	0.1000	Same as above.
"967"	"fSwimHeightScale"	0.9000	Determines how close to the surface you have to be before the breathe indicator goes away
"968"	"fHoldBreathTime"	20.0000	The number of seconds your PC can hold her breath. base time that a character can remain underwater before incurring suffocation damage
"969"	"fHoldBreathEndMult"	0.5000	How Endurance affects the time you can hold your breath. I believe this is a flat-out multiplier to End, added as seconds to HoldBreathTime.<Doesn't seem to work.>
"970"	"fSuffocationDamage"	3.0000	The amount of health damage you take each second you suffocate
"971"	"fMinFlySpeed"	5.0000	Exactly as it says – minimum flying speed.
"972"	"fMaxFlySpeed"	300.0000	See above
"973"	"fStromWindSpeed"	0.7000	UNSURE - Determines altered walk speed during an ash or blight storm, but I'm unsure how – Might be a separate value from that entirely – might determine speed of storm particles /sprites(Dust, etc.) Interesting (possibly related) note, while treading water in an ash storm, I noticed I was moving slightly.
"974"	"fStromWalkMult"	0.2500	Determines altered walk speed during an ash or blight storm, but I'm unsure how uses the getwindspeed to lower the PC movement speed during storms...
"975"	"fFallDamageDistanceMin"	400.0000	The minimum distance you have to fall before you take damage. (Presumably in units) In game units each unit - .056 inches
"976"	"fFallDistanceBase"	0.0000	This will increase/decrease the distance needed to fall before you take damage.
"977"	"fFallDistanceMult"	0.0700	Higher you are the more damage you take when you hit
"978"	"fFallAcroBase"	0.2500	Acrobatics skill increases the distance you can fall before you take damage.
"979"	"fFallAcroMult"	0.0100	Has to do w/ how the Acrobatics skill effects Fall Distance and Damage, but unsure how
"980"	"iMaxActivateDist"	192	Maximum distance for the player to be able to 'Activate' an object - approx 9 feet
"981"	"iMaxInfoDist"	192	Maximum distance for an Info message (object/NPC/creature name, etc.) to pop up in the Player's view
"982"	"fVanityDelay"	30.0000	Seconds until VanityMode begins – the camera starts circling the player if there is no input via mouse or keyboard.
"983"	"fMaxHeadTrackDistance"	400.0000	IIRC, this is the maximum distance an NPC or creature can be away from another NPC or creature and still trigger the 'head follow' routine you sometimes see. Put your PC in Balmora, let it go to Vanity View and you'll see your PC watch passing NPCs and 'follow' their movements for a certain amount of time.
"984"	"fInteriorHeadTrackMult"	0.5000	UNSURE – something to do w/ the modifier for this in Interiors – Do they track at half-distance in Interiors?
"985"	"iHelmWeight"	5	These values are used to set what weights are used to determine whether a piece of armor is light, medium, or heavy. Altering a value alters these categories for *all* armor of that type *everywhere* in the game. Rather nice, actually; I used it in redux to set weight categories for all of my armor types across the board.
"986"	"iPauldronWeight"	10	
"987"	"iCuirassWeight"	30	
"988"	"iGauntletWeight"	5	
"989"	"iGreavesWeight"	15	
"990"	"iBootsWeight"	20	
"991"	"iShieldWeight"	15	
"992"	"fLightMaxMod"	0.6000	These values are used in conjunction with armor weights to set the weight classes (Light, Medium, Heavy).
"993"	"fMedMaxMod"	0.9000	
"994"	"fUnarmoredBase1"	0.1000	
"995"	"fUnarmoredBase2"	0.0650	These two values dictate the range of AR for characters going Unarmored, based on the Unarmored skill – As they are, the settings produce a Maximum Unarmored AR of 65 (at 100 Unarmored skill), which you can see is some for of multiplication between the two settings – Reversing the values produces the same effect , so it seems the values are interchangeable (unless I missed something – could be wrong) - Changing one to 1.000 and the other to 0.0650 results in a max Unarmored AR of 650 – The game multiplies the numbers and then multiplies the resulting number by 1000 to obtain the actual in-game Max Unarmored AR – Doesn't seem to effect Min AR independently, only max - Progression of AR (from low skill to high skill) appears to be hard-coded. It has also been found that the Unarmored skill doesn't work at all UNLESS at least one item of armor is worn. (Forum Info / The other Felix)
"996"	"iBaseArmorSkill"	30	The Skill Level where in-game armors reach their base (i.e. 'In-Editor') AR value – Example: Glass Armor has a Base AR of 50 – At Light Armor skill level 30 (as indicated above), it will read as having AR 50 in-game – Before Skill Level 30, armors have diminished AR's from their Base Value, and after Skill Level 30, armors have higher AR's than their base until Skill Level reaches 100 – I haven't figured out the game's scheme for determining the mult value yet
"997"	"fBlockStillBonus"	1.2500	UNSURE – Presumably the amount that standing still increases the chance to block

"998"	"fDamageStrengthBase"	0.5000	Your STR adds to the damage you do with weapons. This determines how much damage is added.
"999"	"fDamageStrengthMult"	0.1000	Effects amount that Strength effects damage dealt in combat (Unsure of how this value relates to in-game effect)
"1000"	"fSwingBlockBase"	1.0000	
"1001"	"fSwingBlockMult"	1.0000	
"1002"	"fFatigueBase"	1.2500	How much fatigue you lose while walking. However, this appears to let you jump very high without getting hurt (Bug? – Forum Info / DinkumThinkum). 1002 – 1022 All effect 'Fatigue' in-game, obviously – For separate actions, although not all of them actually have an effect in-game (I've never been able to get spells to reduce fatigue) – They seem pretty self-explanatory, but I haven't tested them thoroughly
"1003"	"fFatigueMult"	0.5000	Used to determine successful chance of casting if you are fatigued
"1004"	"fFatigueReturnBase"	2.5000	How much fatigue you regain per second. This is why you don't actually fatigue while walking.
"1005"	"fFatigueReturnMult"	0.0200	How much fatigue returns per second while walking
"1006"	"fEndFatigueMult"	0.0400	
"1007"	"fFatigueAttackBase"	2.0000	How much fatigue you lose with every melee attack you make
"1008"	"fFatigueAttackMult"	0.0000	
"1009"	"fWeaponFatigueMult"	0.2500	
"1010"	"fFatigueBlockBase"	4.0000	How much fatigue you lose blocking with a shield.
"1011"	"fFatigueBlockMult"	0.0000	This will increase the amount of fatigue lost when blocking with a shield.
"1012"	"fWeaponFatigueBlockMult"	1.0000	
"1013"	"fFatigueRunBase"	5.0000	How much fatigue you lose running.
"1014"	"fFatigueRunMult"	2.0000	This one appears to work with encumbrance, the more encumbered the more fatigue you lose/second
"1015"	"fFatigueJumpBase"	5.0000	How much fatigue you lose jumping.
"1016"	"fFatigueJumpMult"	0.0000	Modifier for fatigue loss
"1017"	"fFatigueSwimWalkBase"	2.5000	How much fatigue you lose swimming at a 'walk'
"1018"	"fFatigueSwimRunBase"	7.0000	How much fatigue you lose swimming at a 'run'.
"1019"	"fFatigueSwimWalkMult"	0.0000	Modifier for fatigue loss
"1020"	"fFatigueSwimRunMult"	0.0000	Modifier for fatigue loss
"1021"	"fFatigueSneakBase"	1.5000	The base level of fatigue loss while sneaking
"1022"	"fFatigueSneakMult"	1.5000	Multiplier to that base level
"1023"	"fMinHandToHandMult"	0.1000	
"1024"	"fMaxHandToHandMult"	0.5000	
"1025"	"fHandtoHandHealthPer"	0.1000	
"1026"	"fCombatInvisoMult"	0.2000	Reduce the chance to hit the PC when he is chameleoned or invisible
"1027"	"fCombatKODamageMult"	1.5000	
"1028"	"fCombatCriticalStrikeMult"	4.0000	This one appears to only work if you hit someone unawares while sneaking. I never got it to do anything else. 4x damage from a successful sneak attack works when chameleoned or invisible
"1029"	"iBlockMinChance"	10	Minimum chance of blocking with a shield
"1030"	"iBlockMaxChance"	50	Maximum chance of blocking with a shield
"1031"	"fLevelUpHealthEndMult"	0.1000	Multiplies current END to get hit points added at level-up.
"1032"	"fSoulGemMult"	3.0000	A soul gem's monetary value is multiplied by this value to determine the soul capacity of a soul gem. Creatures with a soul value less than or equal to that capacity can "fit" in the gem.
"1033"	"fEffectCostMult"	0.5000	The setting for all magicka costs for all spell effects. Changing this will change what all spells and enchantments cost. Everything. Linear change. Doubling this makes all spells cost twice as much magicka, all enchanted items cost twice as many charges.
"1034"	"fSpellPriceMult"	2.0000	
"1035"	"fFatigueSpellBase"	0.0000	
"1036"	"fFatigueSpellMult"	0.0000	
"1037"	"fFatigueSpellCostMult"	0.0000	
"1038"	"fPotionStrengthMult"	0.5000	
"1039"	"fPotionT1MagMult"	1.5000	
"1040"	"fPotionT1DurMult"	0.5000	
"1041"	"fPotionMinUsefulDuration"	20.0000	
"1042"	"fPotionT4BaseStrengthMult"	20.0000	
"1043"	"fPotionT4EquipStrengthMult"	12.0000	
"1044"	"fIngredientMult"	1.0000	Min # of an ingredient required to make a potion

"1045"	"fMagicItemCostMult"	1.0000	UNUSED
"1046"	"fMagicItemPriceMult"	1.0000	
"1047"	"fMagicItemOnceMult"	1.0000	
"1048"	"fMagicItemUsedMult"	1.0000	
"1049"	"fMagicItemStrikeMult"	1.0000	
"1050"	"fMagicItemConstantMult"	1.0000	
"1051"	"fEnchantmentMult"	0.1000	is the setting for how much enchantment an item can hold based upon the value set in each individual item's property file. Linear, if an item in TESCS shows an enchantment value of 1200 (i.e. an exquisite ring) multiply it by fEnchantmentMult to get the actual enchantment you'll see in the make an enchanted item window.
"1052"	"fEnchantmentChanceMult"	3.0000	These affect the PC's chance of making an enchantment
"1053"	"fPCbaseMagickaMult"	1.0000	This sets the spell point multiplier for the PC with respect to INT (e.g., 1 x INT with this setting)
"1054"	"fNPCbaseMagickaMult"	2.0000	This does the same thing for NPCs.
"1055"	"fAutoSpellChance"	80.0000	
"1056"	"fAutoPCSpellChance"	50.0000	
"1057"	"iAutoSpellTimesCanCast"	3	
"1058"	"iAutoSpellAttSkillMin"	70	
"1059"	"iAutoSpellAlterationMax"	5	
"1060"	"iAutoSpellConjurationMax"	2	
"1061"	"iAutoSpellDestructionMax"	5	
"1062"	"iAutoSpellIllusionMax"	5	
"1063"	"iAutoSpellMysticismMax"	5	
"1064"	"iAutoSpellRestorationMax"	5	
"1065"	"iAutoPCSpellMax"	100	
"1066"	"iAutoRepFacMod"	2	A positive modification to relations you get with people who belong to the same faction
"1067"	"iAutoRepLevMod"	0	You can apparently add rep points with each level-up. I've never tried it.
"1068"	"iMagicItemChargeOnce"	1	1068-1071 effect the amount of charges auto-calculated on magic items based on their function – This value is the number of uses that the game will account for when calculating the max charges of a magic item (works universally, across the board with all –ingame items – 1068 is the setting for the number of charges an automatically calculated cast once effect enchanted item will have. Formula is BaseSpellEffectCost x iMagicItemChargeOnce. Linear. This way an item with a cast once effect will have exactly the number of charges needed to cast the effect upon it 1069 for const effect items 1070 is the setting for the multiplier for charges for automatically calculated cast when used effect enchanted items. See above for explanation. 1071 for "Cast on Strike" items (charges are calculated to account for X 'uses' with this value as is)
"1069"	"iMagicItemChargeConst"	10	
"1070"	"iMagicItemChargeUse"	5	
"1071"	"iMagicItemChargeStrike"	10	
"1072"	"iMonthsToRespawn"	4	The time to respawn things like the Fighters Guild/Mages Guild chests, etc. How many months before a picked plant respawns ingredients. Chests in guilds respawn contents the same as any other chests.
"1073"	"fCorpseClearDelay"	72.0000	How many hours it takes before a non-persistent corpse disappears.
"1074"	"fCorpseRespawnDelay"	72.0000	How many hours it takes before a respawnable creature actually respawns (note that this doesn't seem to work properly).
"1075"	"fBarterGoldResetDelay"	24.0000	How many hours it takes before a trader resets its barter gold to its default value.
"1076"	"fEncumbranceStrMult"	5.0000	A straight multiplier to STR to see how much a PC/NPC/creature can carry.
"1077"	"fPickLockMult"	-1.0000	Dictates amount that Lock pick difficulty raises according to Lock Level – Lower the value here, the harder it gets – Positive values make locks get easier with higher lock Levels
"1078"	"fTrapCostMult"	0.0000	Dictates difficulty of traps based on the spell cost of the spell assigned as trap – Lower the value here, the harder it gets to disarm (again, based on spell cost of assigned 'trap') The values is multiplied by the spell cost of a trap and then added to your chance of disarming it. Since it's set to zero, the trap spell's cost is not incorporated into the chance. So basically it's also unused.
"1079"	"fMessageTimePerChar"	0.1000	
"1080"	"fMagicItemRechargePerSecond"	0.0500	This is the setting for the amount of charges restored to a charged magic item per second of game play. Linear. 0.05 x 20 seconds = 1 charge restored.
"1081"	"i1stPersonSneakDelta"	10	
"1082"	"iBarterSuccessDisposition"	1	If you barter with a merchant successfully, your disposition with that merchant increases by one and falls by 1 if you fail a barter attempt.
"1083"	"iBarterFailDisposition"	-1	
"1084"	"iLevelupTotal"	10	How many skill points you need before you level up.
"1085"	"iLevelupMajorMult"	1	How much each major skill is worth in points. E.g., if you set this to 2 then each point earned in a major skill counts as 2 skill points for leveling up.

"1086"	"iLevelupMinorMult"	1	Same as above, but for minor skills.
"1087"	"iLevelupMajorMultAttribute"	1	I *think* - not sure if I remember this correctly - but I think this works like the above, but for skills governed by your two primary attributes. So if your primary attributes are STR and AGI, and you set 1087 to 2, then any major skill governed by one of these attributes which goes up by a point counts as 2 points for purposes of leveling.
"1088"	"iLevelupMinorMultAttribute"	1	
"1089"	"iLevelupMiscMultAttribute"	1	
"1090"	"iLevelupSpecialization"	1	The game keeps track of how many skill points you've gained since the last level up. If you gained 8 skill points in skills governed by AGI, then when you get to distribute attribute points whatever number is in place for iLevelUp08Mult will be used for AGI. So if this value is 4, you'll see a x 4 next to AGI when you level up (you'll get 4 points in AGI if you pick this during the leveling process).
"1091"	"iLevelUp01Mult"	2	
"1092"	"iLevelUp02Mult"	2	
"1093"	"iLevelUp03Mult"	2	
"1094"	"iLevelUp04Mult"	2	
"1095"	"iLevelUp05Mult"	3	
"1096"	"iLevelUp06Mult"	3	
"1097"	"iLevelUp07Mult"	3	
"1098"	"iLevelUp08Mult"	4	
"1099"	"iLevelUp09Mult"	4	
"1100"	"iLevelUp10Mult"	5	
"1101"	"iSoulAmountForConstantEffect"	400	This is the setting for the minimum soul value to toggle the constant effect button in the enchantment creation window.
"1102"	"fConstantEffectMult"	15.0000	UNUSED
"1103"	"fEnchantmentConstantDurationMult"	100.0000	This setting is the multiplier for constant effect cast cost as compared to a 0 duration spell. so restore health 2-2 for 0 secs, which costs 0.50 to cast as a spell, costs 0.5 x 100 = 50 as a constant effect.
"1104"	"fEnchantmentConstantChanceMult"	0.5000	
"1105"	"fWeaponDamageMult"	0.1000	weapon damage during combat. depreciation as it were.
"1106"	"fSeriousWoundMult"	0.0000	UNUSED
"1107"	"fKnockDownMult"	0.5000	This sets the chance for a knock-down fActored on how much damage you do in a single blow.
"1108"	"iKnockDownOddsBase"	50	Sets the base odds for a knockdown when the condition for it is met
"1109"	"iKnockDownOddsMult"	50	
"1110"	"fCombatArmorMinMult"	0.2500	
"1111"	"fHandToHandReach"	1.0000	Sets the reach of HTH weapons. Values of less than 1.0 have no meaning.
"1112"	"fVoiceIdleOdds"	10.0000	Controls likelihood of an NPC 'speaking' a voice clip when Idle (Unsure of specifics)
"1113"	"iVoiceAttackOdds"	10	Controls likelihood of an NPC 'speaking' a voice clip when attacking (Unsure of specifics)
"1114"	"iVoiceHitOdds"	30	Controls likelihood of an NPC 'speaking' a voice clip when being hit (Unsure of specifics)
"1115"	"fProjectileMinSpeed"	400.0000	Sets the minimum speed of projectile weapons
"1116"	"fProjectileMaxSpeed"	3000.0000	Dictates maximum speed of projectiles from bows and crossbows
"1117"	"fThrownWeaponMinSpeed"	300.0000	Sets the minimum speed of thrown weapons
"1118"	"fThrownWeaponMaxSpeed"	1000.0000	Dictates Max speed of thrown weapons
"1119"	"fTargetSpellMaxSpeed"	1000.0000	Sets the speed of spells. Double this and your spells will *zip* across the screen! – Min speed is apparently hard-coded
"1120"	"fProjectileThrownStoreChance"	25.0000	The odds of getting arrows back when you loot a corpse. Thrown weapons also
"1121"	"iPickMinChance"	5	UNSURE – Don't know if this is with pick pocketing or lock picking
"1122"	"iPickMaxChance"	75	UNSURE – Don't know if this is with pick pocketing or lock picking
"1123"	"fDispRaceMod"	5.0000	You have better relations with your own race than with others.
"1124"	"fDispPersonalityMult"	0.5000	These determine how personality affect NPC disposition
"1125"	"fDispPersonalityBase"	50.0000	
"1126"	"fDispFactionMod"	3.0000	These determine how your rank in a faction will alter your relations with people that belong to that faction. This is why when you reach high ranks in a faction everyone in that faction suddenly becomes very friendly.
"1127"	"fDispFactionRankBase"	1.0000	
"1128"	"fDispFactionPocketMod"	0.5000	
"1129"	"fDispCrimeMod"	0.0000	This is multiplied by the player's crime level (bounty) to determine how that information affects an NPC's disposition towards the player.
"1130"	"fDispDiseaseMod"	-10.0000	How much disposition is lowered when you're suffering from a disease.
"1131"	"iDispAttackMod"	-50	Not completely sure – NPC disposition modifier if PC attacks said NPC
"1132"	"fDispWeaponDrawn"	-5.0000	How much disposition is lowered when you have a weapon drawn.
"1133"	"fDispBargainSuccessMod"	1.0000	I don't remember if these work the same as the previous barter disposition values, or if these are multipliers. These effect the long term disposition of the merchant
"1134"	"fDispBargainFailMod"	-1.0000	
"1135"	"fDispPickPocketMod"	-25.0000	NPC disposition modifier for catching the PC attempting to pickpocket them
"1136"	"iDaysinPrisonMod"	100	determines prison time based on your crime level.
"1137"	"fDispAttacking"	-10.0000	Unsure – I believe it's an NPC Disposition modifier if the PC is attacking something other than the NPC it effects non-combatants disposition.
"1138"	"fDispStealing"	-0.5000	Unsure - I believe it's an NPC Disposition modifier if the PC is stealing from someone other than the NPC it effects

[Table of Contents](#)
[Table of Contents](#)

"1139"	"iDispTresspass"	-20	NPC Disposition modifier for catching the PC 'trespassing' – Not sure what that exactly means in-game
"1140"	"iDispKilling"	-50	Unsure NPC Disposition modifier for witnessing the PC kill an innocent NPC (I think...)
"1141"	"iTrainingMod"	10	determines training costs. The higher the value, the more training costs. – unsure of method of calculation
"1142"	"iAlchemyMod"	2	
"1143"	"fBargainOfferBase"	50.0000	This is multiplied by the item's value to determine what the merchant will offer when selling. Base value is also modified by PC level. Base amount that merchants will buy items from you for, in percentage points – Believe it goes both ways, but I'm unsure how it would work the 'other way'
"1144"	"fBargainOfferMulti"	-4.0000	Effects how much the merchant lowers his offers during a bargaining session
"1145"	"fDispositionMod"	1.0000	
"1146"	"fPersonalityMod"	5.0000	
"1147"	"fLuckMod"	10.0000	IIRC, this is multiplied by your Luck as a percentage to get a base increase to all skills.
"1148"	"fReputationMod"	1.0000	
"1149"	"fLevelMod"	5.0000	
"1150"	"fBribe10Mod"	35.0000	Dictates amount that NPC Disposition will raise on a successful 10 Gold Bribe – Don't believe it's in straight disposition points – Could be percentages - (Other fFactors like race, sex, opposing faction etc. reduce this amount significantly)
"1151"	"fBribe100Mod"	75.0000	Dictates amount that NPC Disposition will raise on a successful 100 Gold Bribe. See above.
"1152"	"fBribe1000Mod"	150.0000	Dictates amount that NPC Disposition will raise on a successful 1000 Gold Bribe. See above.
"1153"	"fPerDieRollMult"	0.3000	
"1154"	"fPerTempMult"	1.0000	is used in just about every disposition modifying calculation.
"1155"	"iPerMinChance"	5	
"1156"	"iPerMinChange"	10	
"1157"	"fSpecialSkillBonus"	0.8000	These all determine how fast you gain skill points in each skill. The lower the value, the faster you'll gain skill points. The values are multiplied by whatever rate you set for each individual skill.
"1158"	"fMajorSkillBonus"	0.7500	
"1159"	"fMinorSkillBonus"	1.0000	
"1160"	"fMiscSkillBonus"	1.2500	
"1161"	"iAlarmKilling"	90	
"1162"	"iAlarmAttack"	50	
"1163"	"iAlarmStealing"	1	
"1164"	"iAlarmPickPocket"	20	
"1165"	"iAlarmTresspass"	5	
"1166"	"fAlarmRadius"	2000.0000	When an NPC raises the alarm, this is the base radius for response by other affiliated NPCs.
"1167"	"iCrimeKilling"	1000	These set the gold value for crimes. I believe that fCrimeStealing is multiplied by the price of the item stolen.
"1168"	"iCrimeAttack"	40	
"1169"	"fCrimeStealing"	1.0000	
"1170"	"iCrimePickPocket"	25	
"1171"	"iCrimeTresspass"	5	
"1172"	"iCrimeThreshold"	1000	When NPC's start to react negatively to the PC
"1173"	"iCrimeThresholdMultiplier"	10	
"1174"	"fCrimeGoldDiscountMult"	0.5000	Thieves guild discount when you have a price on your head.
"1175"	"fCrimeGoldTurnInMult"	0.9000	Discount on the fine if you turn yourself in.
"1176"	"iFightAttack"	100	
"1177"	"iFightAttacking"	50	
"1178"	"iFightDistanceBase"	20	
"1179"	"fFightDistanceMultiplier"	0.0050	
"1180"	"iFightAlarmMult"	1	
"1181"	"fFightDispMult"	0.2000	
"1182"	"fFightStealing"	50.0000	
"1183"	"iFightPickpocket"	25	
"1184"	"iFightTresspass"	25	
"1185"	"iFightKilling"	50	
"1186"	"iFlee"	0	UNUSED
"1187"	"iGreetDistanceMultiplier"	6	Used for those annoying voice greetings NPCs use when you get too close Specifically (if the construction set help is to be believed) this is multiplied by their hello rating to get the distance before they talk.
"1188"	"iGreetDuration"	4	
"1189"	"fGreetDistanceReset"	512.0000	How far away from an NPC you have to get before they check for a voice greeting again.
"1190"	"fIdleChanceMultiplier"	0.7500	Probability multiplier that an NPC will mumble something while standing idly near the PC

"1191"	"fSneakUseDist"	500.0000	Helps determine if you can sneak
"1192"	"fSneakUseDelay"	1.0000	Helps determine how long before the Sneak Icon come on
"1193"	"fSneakDistanceBase"	0.5000	see above
"1194"	"fSneakDistanceMultiplier"	0.0020	see above
"1195"	"fSneakSpeedMultiplier"	0.7500	Multiplied by base walking speed to see how fast you move while sneaking.
"1196"	"fSneakViewMult"	1.5000	Makes it more difficult to sneak when in view of an NPC.
"1197"	"fSneakNoViewMult"	0.5000	Makes it easier to sneak when you aren't in view.
"1198"	"fSneakSkillMult"	1.0000	
"1199"	"fSneakBootMult"	-1.0000	Multiplied by the boot value (weight?) to determine the reduction to Sneak skill.
"1200"	"fCombatDistance"	128.0000	Combined with weapon reach, determines the effective distance that hits can be obtained
"1201"	"fCombatAngleXY"	60.0000	
"1202"	"fCombatAngleZ"	60.0000	
"1203"	"fCombatForceSideAngle"	30.0000	
"1204"	"fCombatTorsoSideAngle"	45.0000	
"1205"	"fCombatTorsoStartPercent"	0.3000	
"1206"	"fCombatTorsoStopPercent"	0.8000	
"1207"	"fCombatBlockLeftAngle"	-90.0000	Shields are worn on the left and partially block attacks from 90 degrees left to 30 degrees right of the PC's facing.
"1208"	"fCombatBlockRightAngle"	30.0000	
"1209"	"fCombatDelayCreature"	0.1000	
"1210"	"fCombatDelayNPC"	0.1000	
"1212"	"fAI MeleeWeaponMult"	2.0000	Used in the determination of how far away an NPC will flee if they flee combat and the PC has a melee weapon in hand
"1213"	"fAIRangeMeleeWeaponMult"	5.0000	as above but the PC has a crossbow or Bow in hand
"1214"	"fAIMagicSpellMult"	3.0000	
"1215"	"fAIRangeMagicSpellMult"	5.0000	As above but the PC has a spell readied
"1216"	"fAIMeleeArmorMult"	1.0000	
"1217"	"fAIMeleeSummWeaponMult"	1.0000	
"1218"	"fAIFleeHealthMult"	7.0000	Alters the opponents flee rating when health declines
"1219"	"fAIFleeFleeMult"	0.3000	Used to alter base flee ratings.
"1220"	"fPickPocketMod"	0.3000	
"1221"	"fSleepRandMod"	0.2500	Affects the chance of a mob waking the PC up while asleep in the wilderness.
"1222"	"fSleepRestMod"	0.3000	Unused (Thanks to Damar Stiehl for these two)
"1223"	"iNumberCreatures"	1	
"1224"	"fAudioDefaultMinDistance"	5.0000	
"1225"	"fAudioDefaultMaxDistance"	40.0000	
"1226"	"fAudioVoiceDefaultMinDistance"	10.0000	
"1227"	"fAudioVoiceDefaultMaxDistance"	60.0000	
"1228"	"fAudioMinDistanceMult"	20.0000	
"1229"	"fAudioMaxDistanceMult"	50.0000	
"1230"	"fNPCHealthBarTime"	3.0000	Controls delay before the Opponents health bar disappears
"1231"	"fNPCHealthBarFade"	0.5000	Controls how many seconds the bar "fades" for (rather than abruptly vanishing)
"1232"	"fDifficultyMult"	5.0000	
"1399"	"fMagicDetectRefreshRate"	0.0167	
"1400"	"fMagicStartIconBlink"	3.0000	The number of seconds a spell icon will fade before the spell runs out, on the lower right-hand corner of the screen.
"1401"	"fMagicCreatureCastDelay"	1.5000	
"1431"	"fDiseaseXferChance"	2.5000	The chance of catching a disease if hit by a creature, or looting a diseased creature's corpse.
"1432"	"fElementalShieldMult"	0.1000	
"1435"	"fMagicSunBlockedMult"	0.5000	Vampire weakness
	"fWereWolfRunMult"	1.3000	Werewolf run speed multiplier.
	"fWereWolfSilverWeaponDamageMult"	2.0000	The damage multiplier for silver weapon damage against all werewolves.

Morrowind Mod Maker's Manual: Appendix B

	"iWereWolfBounty"	1000	These are the skills and attributes for Werewolf form.
	"fWereWolfStrength"	150.0000	
	"fWereWolfAgility"	150.0000	
	"fWereWolfEndurance"	150.0000	
	"fWereWolfSpeed"	90.0000	
	"fWereWolfHandtoHand"	100.0000	
	"fWereWolfUnarmored"	100.0000	
	"fWereWolfAthletics"	50.0000	
	"fWereWolfAcrobatics"	80.0000	
	"fWereWolfIntelligence"	0.0000	
	"fWereWolfWillPower"	0.0000	
	"fWereWolfPersonality"	0.0000	
	"fWereWolfLuck"	25.0000	
	"fWereWolfBlock"	0.0000	
	"fWereWolfArmorer"	0.0000	
	"fWereWolfMediumArmor"	0.0000	
	"fWereWolfHeavyArmor"	0.0000	
	"fWereWolfBluntWeapon"	0.0000	
	"fWereWolfLongBlade"	0.0000	
	"fWereWolfAxe"	0.0000	
	"fWereWolfSpear"	0.0000	
	"fWereWolfDestruction"	0.0000	
	"fWereWolfAlteration"	0.0000	
	"fWereWolfIllusion"	0.0000	
	"fWereWolfConjuration"	0.0000	
	"fWereWolfMysticism"	0.0000	
	"fWereWolfRestoration"	0.0000	
	"fWereWolfEnchant"	0.0000	
	"fWereWolfAlchemy"	0.0000	
	"fWereWolfSecurity"	0.0000	
	"fWereWolfSneak"	95.0000	
	"fWereWolfLightArmor"	0.0000	
	"fWereWolfShortBlade"	0.0000	
	"fWereWolfMarksman"	0.0000	
	"fWereWolfSpeechcraft"	0.0000	
	"iWereWolfLevelToAttack"	20	
	"iWereWolfFightMod"	100	
	"iWereWolfFleeMod"	100	
	"fWereWolfHealth"	2.0000	
	"fWereWolfFatigue"	400.0000	
	"fWereWolfMagica"	100.0000	
	"fCombatDistaceWereWolfMod"	0.3000	Determines the attack range of a Werewolf.
	"fFleeDistance"	3000.0000	Determines how far away someone will flee.

Appendix C Websites

Here I provide a listing of websites I have found. Some are extremely useful sources of information, others are just great to visit. I provide these with the understanding that the website authors consent to have their sites listed in this manual (why have a website if you don't want anyone to visit it?). I list these in no particular order. Make your own judgements as to which sites are worth visiting. There is also a MW webring, several of those listed below belong to it.

The Official Morrowind Website - <http://www.elderscrolls.com>

A Dream of Asgard - <http://www.asgard.nahallan.com>

Academy for Dwemer Studies - <http://dwemer.whirlingschool.net/index.html>

Adul's Morrowind Pages - <http://www.adul.net/>

Aldrien's Chalice - <http://www.rpgplanet.com/morrowind/chalice>

Argent's Mod Page - <http://members.optusnet.com.au/argent2/mw>

B.E. Griffith (Rhedd's Site) - <http://www.netten.net/~bgriff/index.html>

Baratheon79's Morrowind Mods - <http://morrowind.rpgmods.com/>

Bear371's site - <http://www.bear371.com/>

Bethaliz's Morrowind Plug-ins - <http://www.geocities.com/bumgirl28/index.html>

Brash's MW site - <http://brash.iwebland.com>

Cait Sith - <http://www.caitsith.it/>

Caladan Brood - <http://groups.msn.com/CaladanBrood/moonsspawnbeta.msnw?pgmarket=en-us>

Calislahn's Morrowind Mods - <http://www.calislahnsmods.co.uk/index.htm>

Canadian Ice & Howndog's website - <http://canadianice.ufrealms.net/Morrowind/Index.html>

Carnajo's Mod Page - <http://home.tiscali.co.za/~31003603/index.html>

City of Black Light Homepage - <http://users.sisna.com/cquinn>

Clone Gaming Studios - <http://www.rpgplanet.com/morrowind/sodp/>

Daduke's Site - <http://daduke.m1a1.net>

Darliandor's Alchemy Lab - <http://morrowind.melian.cc/>

Destination Morrowind - <http://morrowind.ttlg.com/>

Dragonsong's Tutorials - <http://www.rpgplanet.com/morrowind/tamriel/tutorials/dragonson%20tuts/index.htm>

Duncan's Plug-In Page - <http://home.wi.rr.com/cairnterrier/>

Elder Scrolls: Kingdoms - <http://www.rpgplanet.com/morrowind/daggerfall/index.html>

Elric Melnibone in Morrowind - <http://www.elricm.com>

Emma's Morrowind Site - <http://lovkullen.net/Emma/index.htm>

Endhome - <http://www.endhome.com/continent.html>

Final Fantasy 7 mod - <http://lotysoftware.free.fr/FF7-reborn/index.htm>

Funky Bob's Mods for Morrowind - <http://morrowindmods.freewebspace.com>

GameFaqs.com - <http://www.gamefaqs.com>

Ghostwheel's utilities - <http://www34.brinkster.com/ghostwheel>

Hannah's Whereizit Morrowind Page - <http://www.buttersky.com/morrowind/index.html>

Heph's Morrowind Page - <http://www.heph.org/morrowind>

Hiredgoon's MW Gateway - <http://www.hiredgoons.net/MWFiles>

[Table of Contents](#)

[Table of Contents](#)

Ichinin's Morrowind Stuff - <http://www.geocities.com/ichinin/mwstuff.htm>
IggyEGuana's website - <http://www.geocities.com/iggyeguana>
Indigo's Vault - <http://www.rpgplanet.com/morrowind/vault>
Jeremy's Knights of Tamriel - http://home.comcast.net/%7Ehdoan0/Knights_of_Tamriel.htm
Katana's Pages - <http://katana.3dgladiators.com/Morrowind/index.html>
Kathykitten's Tutorials - <http://www.angelfire.com/wizard/kathykitten/journeys.html>
Karstux's MW Sanctuary - <http://www.apoapsis.net/>
Khalazza Production Mods - <http://www.khalazzaprod.fr.st>
Lady Eternity's MW Page - <http://www.proudfootimaging.com/ladye>
Lady Moiraine's Morrowind site - <http://www.ladymoiraine.com>
MadMax's World of Scripting Magic - http://www.highpressure-morrowind.com/Madmax_Morrowind.html
Magic Nakor's Morrowind Mods - <http://www.rpgplanet.com/morrowind/nakor>
Mantodea - <http://mantodea.tygirwulf.com/morrowind.html>
MarcusX Morrowind Mods - <http://www.marcusx.com/morrowind>
Martin's Portfolio - <http://www.angelfire.com/co/juicytriangle>
Mental Elf's Morrowind Page - <http://www.aloha.net/%7Efrann/rsg>
MistyMoon Web Site - <http://dizzy.at/mistymoon>
ModMan's Morrowind Site - <http://www.freewebs.com/xura/index.htm>
Morrowind 4 Kids - <http://morrowind4kids.com>
Morrowind Abodes - <http://mindhook.com/morrowindabodes>
Morrowind After Dark - <http://www.rpgplanet.com/morrowind/afterdark>
Morrowind Corner - <http://www.morrowindcorner.com>
Morrowind Creations (Nomad's Site) - <http://www.mwcreations.00freehost.com>
Morrowind Italia - <http://www.rpgplanet.com/morrowind/italia>
Morrowind Mods and Modder Resources - <http://www.morrowind-mod.com>
Morrowind Mods and Models - http://veet_vojagig.tripod.com/concept/Concept.html
Morrowind Mods by Dongle - <http://www.deffeyes.com/morrowind/>
Morrowind Mythic Mods - <http://www.mwmythicmods.com>
Morrowind Source - <http://www.modlibrary.co.uk>
Morrowind Summit - <http://www.rpgplanet.com/morrowind>
Morrowind Visions (Silaria) - <http://www.morrowindvisions.com>
Morrowind Voice Add-on Project - <http://members.home.nl/tomsnellen>
Morrowind Script Extender link page - <http://home1.gte.net/cdcooley>
Mykul's Morrowind Page - <http://mykul.heph.org>
Psychodog Studios (The Better Bodies Project) - <http://bb.psychodogstudios.net>
Qwert's Morrowind Mods - <http://qwerts.mods.web1000.com>
Rethan-Manor - <http://www.rethan-manor.net>
RPG Gaming Gone Wrong - <http://www.rpg-gaming-gone-wrong.com>
Sabregirl's Morrowind Mods - <http://sabregirl.com/>
Sheikizza's mods - <http://www.sheikizza.boneflower.com>
Silgrad Tower - <http://www.silgrad.com/silgrad/index.html>
Silveri's mods - <http://www.silveri.dk/morrowind.html>
Slot's Hive - <http://slofshive.caesium2.com>
Smite's Plight - <http://oregonstate.edu/~virdeb/Index.htm>
Srikandi's Morrowind - <http://members.cox.net/srikandi>

[Table of Contents](#)

TESPCD (Utility) page - <http://www.sovanto.com/morrowind/tespcd.htm>
Texture Freak's Morrowind Store - <http://home.planet.nl/%7Ehardbol>
Thanos Tower - <http://66.106.110.151/Morrowind/index.html>
TheLys's website - <http://www.thelys.org/index.php>
The Black Mill mod - <http://www.angelfire.com/realms3/blackmill>
The Glade (RedwoodTreeSprite's site) - <http://www.zyworld.com/redwoodtreesprite/Home.htm>
The Imperial Library - <http://til.gamingsource.net>
The Tamriel Rebuilt Project - <http://rpgplanet.com/morrowind%5Ctamriel>
The Unforgotten Realms - <http://www.ufrealms.net>
The Unofficial Elder Scrolls Pages - <http://www.uesp.net/index.shtml>
Thommy Khajiit's Hideout - <http://www.tommyshideout.nl>
Titans of Ether (Ultima remake) - <http://ultima.cfkasper.de>
Trylobit's Website - <http://home.hccnet.nl/g.wubben>
Tweak Town website (some help in tweaking your game) -
<http://www.tweaktown.com/document.php?dId=316&dType=guide>
Vanhikes Morrowind Mods - <http://www.angelfire.com/games5/vanhikes>
Venymora's Site - <http://fantasyfreak.netfirms.com/>
Whirling School of Vivec - <http://www.whirlingschool.net/>
Wizard's Islands - <http://wizards-islands.com>
Zdim's Morrowind Mods - <http://www.digital-eel.com/mw/mods>

A Story (I laughed so hard I choked) - <http://machall.com/morrowind>

The following sites are places you can download archive utilities. I take no responsibility for anyone using these, it is up to you if you wish to download and make use of any of these utilities:

7Zip - <http://www.7-zip.org>
WinRar - <http://www.rarsoft.com>
UltimateZip - <http://www.ultimatezip.com>
WinAce - <http://www.winace.com>
ZipGenius - <http://www.zipgenius.it>
WinZip - <http://www.winzip.com>
PKWare - <http://www.pkware.com>

Appendix D Meshes and Textures

Every object in the game that can be placed in the game world has a [Mesh](#) and [Texture](#) associated with it. When creating a new object in the CS, it must be based upon a mesh to be given physical form in the game. Textures are wrapped around the mesh to give it a 'skin', thus making it look like an object. Textures are associated with a specific mesh (usually assigned by the program that created the mesh, though some utilities can do this too). `.dds` files are preferred because they compress to smaller files well, support alpha channels, and still look pretty good. `.tga` files are larger, but have good resolution, and also support alpha channels. `.bmp` files work, but are large, and the least preferred to work with. The easiest way to create a new object is to just use a mesh provided with the game. These can be found on the TESCS CD (for Morrowind), and also the Tribunal and/or Bloodmoon disc (for those objects that come with the appropriate expansion). But what if you don't see a mesh you like? Or maybe you like the shape, but not the coloring? You want to make your own, right? No problem, there are programs you can use to do this. But I hope you have your wallet handy, you may need it.

3D Studio Max.

This software program was used to create the 3D objects and environment within Morrowind. This is expensive professional 3D-model creation software (price was around \$3000, students could get it cheaper). It is made by a company called [Discreet](#). The version you need to use is either v3.0, v4.0, or v5.0. Later versions will not create a `.nif` file that Morrowind can use (currently, v7.0 is available). This software requires a plug-in so that it can import/export `.nif` files (3D Studio Max creates it's own files, which must be converted, or exported, into the `.nif` file format for use by Morrowind). The export plug-in for 3D Studio Max 3 and 4 can be found at the Morrowind official site under Downloads. Most likely you'll have to look through bargain bins at stores and websites to try and find an older copy of 3D Studio Max. I don't believe Discreet offers older versions for sale any more. Import plug-ins can be found at [Thanos Tower](#) for the game. You can just download these.

***Note:** Version 5.0 will enable you to create objects, but the `.nif` exporter has issues apparently. Some people report that some features created with this version do not export properly to a `.nif` file (but you can still use it to create meshes). I cannot verify this as I do not own any version of 3D Studio Max.*

GMax.

GMax is another 3d modeling software program, and it's free (get it from Discreet's website also). Notice I said free. This software was created so modders can create objects for use in certain games. Morrowind is not one of those games. What you can do with this program is create an object, then find someone who has an appropriate version of 3D Studio Max, then beg and plead nicely asking them to convert the file into a `.nif` for you, then send it back to you (if someone does this, be very, very grateful). You will not be able to do any animations, nor anything advanced. But you can create simple objects that can be placed in the game world and seen. You cannot make objects that can be equipped by the PC or actors (that has to be done in 3D Studio Max). So while this is a great little free program, be aware of it's limitations.

[Table of Contents](#)

[Table of Contents](#)

Milkshape.

This is a program that can be used to create 3D models by [Chumbalum Soft](#). You can use it to create meshes for Morrowind, but keep in mind, it wasn't made for Morrowind. Results may vary based upon what you are creating with it. There is a .nif exporter, get it from [NetImmerse File: Liberation Association](#) (NIFLA). The exporter is still in development, but at this time it cannot physique some objects, do transparency, animations, or particle effects. Again, I do not have nor use this software, so cannot verify exactly how useful it is, or any limitations inherent in using it.

PhotoShop.

PhotoShop is a program made by [Adobe](#). It can be used to [retexture](#), or skin an image file (or create a new one). It seems to be quite popular, and quite expensive. You can use PhotoShop to create .bmp, .tga, or .dds files. To create .dds files you must get a plug-in with which you will be able to export your file to the .dds format. These are available at the Morrowind official site.

PaintShopPro.

PaintShopPro is a program originally made by Jasc, but has been acquired by [Corel](#). It too can be used to retexture an image. PaintShopPro by all accounts works just as well as PhotoShop, but is far cheaper. Reportedly, the .dds exporter for PhotoShop also works fine with PaintShopPro. However, that was with prior versions of the program, I cannot verify if it still works with current versions.

TheGimp.

[The Gimp](#) is an image editing program that can also be used to retexture image files. It is a free program. As I am not conversant with using this program, I suggest you look at this online manual called [Grokking The Gimp](#) on how to use it.

Irfanview.

This is a free photo imaging program, named after the author. With it you can view most file formats. If you are savvy enough to download and install the plug-ins from the website, you will find you can open and view .dds files. While you cannot work with or save files in the .dds format, you can save them to another format for altering. Nice. [Irfanview](#), while not able to slice bread, can be very useful for those among us in the lower tax brackets.

MWGraphics.

You can also try your luck with [MWGraphics](#) (no, the MW does not stand for Morrowind, it's the site authors initials). This site has some tools that can enable you to change image files to other formats, such as .dds. Make sure you get all the files needed to make use of the utilities here.

Appendix E Contacting

I would greatly appreciate any comments on this work, even negative. If you wish to do so, email me. I will make every effort to reply as soon as possible. I can also be contacted at the several websites, my nom de pleur is Edwardmd. At home, I'm just simply dad. When I tick off my wife, its mud.

Email comments and feedback to: edwardsmd@gmail.com

Good luck and happy modding!

Index

I have tried to be thorough, though no doubt I have missed a few references.

.ace.....	VIII, 9
.bmp.....	VIII, X, 3, 4, 10, 12, 136, 176, 177
.dds.....	VIII, 3, 10, 136, 141, 176, 177
.ini.....	3, 4, 5, 12, 91, 147
.nif.....	IX, 3, 10, 136, 142, 176, 177
.rar.....	IX, 9, 175
.tga.....	X, 3, 10, 36, 136, 176, 177
.zip.....	VIII, IX, X, 9, 10, 175
3D Studio Max.....	VIII, IX, 1, 176
Ability.....	41, 46, 50, 51, 56
Activator.....	3, 35, 138, 146
Active File, Active Plug-in.....	1, 18, 19, 137
Adobe.....	VIII, IX, 58, 177
Adobe Photoshop.....	see Adobe
AI.....	13, 15, 41, 47, 62, 88-90, 101, 103, 146, 150, 151, 161, 162, 179, 171
Alchemy.....	35, 43, 98, 114, 123, 138, 164, 169, 171, 173
AllowWereWolfForceGreeting.....	127
Angle Snap.....	20, 30, 53
Animations.....	3, 28, 40, 41, 42, 47, 54, 56, 112, 117, 119, 125, 156, 158, 159, 162, 176, 177
Apparatus.....	35, 138
Armor.....	36-39, 47-49, 70, 87, 89, 97, 98, 101, 102, 114, 115, 123, 138, 149, 150, 164, 166, 169
Armor Type Weight Allocation Table.....	37
Attributes.....	VIII, IX, 1, 2, 15, 41, 42, 46, 50, 55, 57, 63, 64, 70, 145, 149, 164, 168, 171
Auto-calculate.....	40, 43, 46, 51, 167
Axis (x, y, z), -plane.....	X, 24, 53, 59, 61, 62, 101, 107, 109, 115, 116, 122-125, 150, 154, 147, 161
Bethesda.....	2, 4, 6, 12, 26, 61, 65, 66, 74-76, 79, 141, 145
Biped.....	37, 42, 56
Birthsigns.....	28, 33, 50, 58, 93, 95, 130, 138
Blood texture (color).....	41, 46
Bloodmoon.....	VIII, 2, 6-8, 11, 25, 29, 69, 73, 75, 90, 106, 107, 110, 111, 135, 126 135, 142, 176
Body Part.....	36-40, 42, 46, 101, 102, 138
Book.....	39, 87, 129, 135, 138, 141
Brightness.....	X, 23, 31, 45, 52
Buffer Cells.....	20, 60, 125
Bugs.....	15, 136
Cell Loading.....	20, 41, 43, 46, 49, 60, 62

[Table of Contents](#)

[Table of Contents](#)

Cells.....	5, 12, 13, 16, 19-27, 30, 33, 41, 43, 46, 48, 49, 52-54, 59-62, 67, 69, 72, 75, 81, 87-91, 97, 105-108, 114, 118-120, 125, 130-132, 135, 138, 139, 146, 147, 151, 152, 156, 159, 160, 162, 163, 179
Character Generation.....	130, 131
Clipping Distance Slider.....	21, 143
Class.....	28, 46, 57, 58, 66-69, 72, 75, 76, 94, 95, 114, 130, 138, 150, 156
Clothing.....	36, 38, 39, 70, 76, 89, 138
Collision.....	X, 14, 23, 162, 163
Combat.....	14, 15, 41, 48, 57, 69, 74, 109, 125, 147, 162, 163, 166-170, 172
Combining Plug-ins.....	19
Commands.....	11-15, 74, 78, 79, 82-84, 86, 146
Comma.....	79, 113, 114
Compile Scripts.....	31, 79, 80, 83
Compressed Files.....	VIII, IX, X, 3, 9, 10, 136, 176
Console.....	3, 11-15, 33, 78, 91, 97, 124, 129, 147, 155, 162, 163
Contacting Info.....	178
Container.....	40, 44, 45, 87, 105, 138, 146
Corpses Persist.....	41, 46, 168, 169, 171
Crash To Desktop, CTD.....	VIII, 4, 89, 91, 92, 121, 130
CreateMaps.....	12, 147
Creatures.....	IX, 14, 26, 27, 29, 40-46, 50, 51, 62, 63, 68-70, 72, 73, 75, 87-90, 93, 103, 110, 121, 134, 138, 143, 145, 146, 148, 155, 160, 161, 163, 165, 167-168, 170, 171
Curse.....	1, 49, 51, 87, 88, 108, 121, 145, 146, 154, 161
Data Files.....	VIII, IX, 1-3, 6, 10, 11, 18, 19, 21, 30, 35-40, 42-45, 49, 51, 137, 139
Day (global).....	127, 147
DaysPassed.....	127, 147
Debug.....	15, 26, 137
Dialog Conflicts.....	21
Delete.....	2, 8, 10, 15, 26, 27, 29, 31, 34, 42, 47, 49, 52, 55-58, 63-65, 124, 130, 131-134, 136, 138, 139, 143, 162
Dependency.....	VIII, 5-8, 18, 19, 67, 135, 142
Detail button.....	3, 19, 31, 137, 139
Dialog.....	5, 6, 13, 19, 21, 22, 27, 28, 31, 33, 41, 47, 48, 63-78, 81, 88, 90, 97, 136, 138, 139, 142, 146-148, 155, 163
Difficulty Slider.....	143
Dirty Mod.....	2, 16, 19, 34, 66, 135, 137, 141, 142
Discreet.....	VIII, 176
Disease.....	50, 51, 70, 77, 87, 88, 90, 102, 103, 121, 146, 151, 154, 161, 169, 171
Disposition.....	46, 47, 69, 76, 98, 99, 115, 123, 149, 168, 169
Door.....	23, 27, 42, 75, 87, 88, 138
Doubling.....	136, 137
Duplicate.....	VIII, 22, 53
Edit.....	21, 22, 27, 29-31, 34, 55, 56, 59-62, 131, 137
Enchant.....	37, 39, 41-43, 48, 50, 51, 58, 98, 115, 123, 128, 133, 138, 145, 164, 167, 168, 171
Encumbrance.....	40, 42, 121, 165, 166, 168

Error Messages.....	4, 6, 18, 73, 80
ESM.....	VIII, 1-7, 13, 17-19, 21, 34, 65, 124, 135, 139
ESP.....	VIII, 1-13, 17-19, 34, 139
ESS.....	VIII, 2, 3, 139
Essential.....	40, 46
Expansions.....	see Tribunal or Bloodmoon
Exterior.....	12-15, 20, 24, 25, 27, 53, 71, 73, 81, 88, 91, 105, 106, 108, 118, 120, 125, 130, 143, 146, 147, 159, 160, 162, 163
Fatigue.....	VII, 14, 41, 46, 50, 70, 98, 115, 123, 145, 149, 157, 166, 167, 172
FillJournal.....	12, 148
Find.....	22
FixMe.....	97, 148
Fog, Foggy.....	14, 23, 24, 31, 71, 91, 103, 115, 125, 147, 151, 157, 163
Fog Of War.....	14, 125, 163
ForceJump.....	97, 104, 149
ForceMoveJump.....	97, 104, 149
ForceRun.....	97, 104, 149
ForceSneak.....	97, 104, 149
Forums.....	8, 11, 17, 136, 144
Frames.....	VIII, 4, 81, 124, 162
Frames Per Second, FPS.....	VIII, 4, 11, 29, 52, 84, 124, 162
Functions.....	1, 2, 11, 12, 18, 65-67, 69-71, 73-81, 83-132, 145-172
Game Settings.....	VIII, 6-8, 28, 125, 138, 143, 164-172
GameHour.....	103, 127, 149
GenMods.....	141
Get-.....	97-101
-Alarm.....	98, 148
-ArmorBonus.....	98, 148
-AttackBonus.....	98, 148
-Attributes.....	98, 148
-Blindness.....	99, 148
-CastPenalty.....	99, 148
-Chameleon.....	99, 148
-DefendBonus.....	99, 148
-Disposition.....	99, 148
-Fight.....	99, 148
-Flee.....	99, 148
-Flying.....	100, 148
-Hello.....	100, 148
-Invisible.....	100, 149
-Level.....	149
-Paralysis.....	100, 148
-PCCrimeLevel.....	100, 148
-PCVisionBonus.....	100, 148
-Reputation.....	100, 148
-Resist.....	148

[Table of Contents](#)

-Scale.....	100, 149
-Silence.....	101, 148
-Skills.....	148
-SuperJump.....	109, 148
-SwimSpeed.....	101, 148
-WaterBreathing.....	109, 148
-WaterLevel.....	101, 149
-WaterWalking.....	109, 148
GetAIPackageDone.....	101, 149
GetAngle.....	101, 149
GetArmorType.....	101, 149
GetAttacked.....	102, 150
GetBlightDisease.....	102, 150
GetButtonPressed.....	102, 113, 150
GetCollidingActor.....	102, 150
GetCollidingPC.....	102, 150
GetCommonDisease.....	102, 150
GetCurrentAIPackage.....	102, 150
GetCurrentTime.....	103
GetCurrentWather.....	103, 150
GetDeadCount.....	103, 150
GetDetected.....	103, 150
GetDisabled.....	104, 151
GetDistance.....	97, 104, 151
GetEffect.....	104, 151
GetFactionReaction.....	99, 149
GetForceJump.....	104, 148
GetForceMoveJump.....	104, 148
GetForceRun.....	104, 148
GetForceSneak.....	104, 148
GetHealthGetRatio.....	104, 151
GetInterior.....	105, 151
GetItemCount.....	105, 121, 151
GetJournalIndex.....	105, 149
GetLineOfSight.....	105, 151
GetLocked.....	105, 151
GetLOS.....	105, 151
GetMasserPhase.....	105, 151
GetPCCell.....	106, 151
GetPCFacRep.....	100, 149
GetPCJumping.....	106, 152
GetPCRank.....	106, 152
GetPCSleep.....	106, 152
GetPCTraveling.....	107
GetPlayerControlsDisabled.....	107, 152

[Table of Contents](#)

[Table of Contents](#)

GetPlayerFightingDisabled.....	107, 152
GetPlayerJumpingDisabled.....	107, 152
GetPlayerLookingDisabled.....	107, 152
GetPlayerMagicDisabled.....	107, 152
GetVanityModeDisabled.....	107, 153, 154
GetPlayerViewSwitch.....	107, 153
GetPos.....	107, 149
GetRace.....	107, 153
GetSecondsPassed.....	103, 107, 153
GetSecundaPhase.....	107, 153
GetPCSneaking.....	106, 152
GetSoundPlaying.....	108, 153
GetSpell.....	108, 153
GetSpelleffects.....	108, 153
GetSpellReadied.....	108, 153
GetSquareRoot.....	108, 153
GetStandingActor.....	108, 153
GetStandingPC.....	108, 153
GetStartingPos.....	109, 153
GetTarget.....	109, 154
GetWeaponDrawn.....	109, 154
GetWeaponType.....	109, 154
GetWerewolfKills.....	110, 154
GetWindSpeed.....	110, 154
Global.....	13, 17, 29, 54, 69, 72, 81, 82, 85, 90, 113, 121, 125-129, 144, 146 155-158, 161, 163
GMST.....	VIII, 6-8, 124, 138
Go To Cell.....	26
GoToJail.....	110, 154
Granularity.....	62
Grid Snap.....	20, 30, 53
Guilds.....	1, 167
HasSoulGem.....	110, 154
Health.....	14, 37, 41, 46, 50, 51, 70, 98, 104, 105, 114, 115, 122, 144, 148, 151 155, 156, 163, 164, 166, 167, 170, 171
Help (console).....	13, 14, 17, 29, 154, 163
HitAttemptOnMe.....	110, 155
HitOnMe.....	110, 155
HurtCollidingActor.....	111, 155
Import/Export Data.....	16, 21
Ingredient.....	43, 138, 163, 166, 167
Installing.....	9-11, 17, 18, 135, 142
Interior.....	13, 20, 24, 25, 27, 49, 52, 53, 56, 75, 88-90, 105-107, 118, 120, 129, 135 136, 140, 151, 153, 158, 159, 161, 162, 165
Inventory.....	35-37, 39, 40, 42-45, 47-49, 51, 67, 69, 72, 78, 87, 89, 94, 96, 108, 110, 116, 120, 121, 127, 128, 130, 145, 147, 151, 153, 154, 156, 160
Isometric View.....	23

IsWerewolf.....	111, 155
Journal.....	12, 66-69, 72, 73, 76-78, 105, 111, 124, 147, 149, 155
Journal (function).....	111, 155
Land Sensitivity Multiplier.....	20, 60
Landscape.....	X, 14, 20, 23, 27, 30, 52, 59-62, 138
Leveled Creatures.....	43, 44, 88, 120, 138
Leveled Items.....	42, 44, 120, 138
Leveled List.....	IX, 11, 26, 40, 88, 120, 145, 159
Leveled List Merger.....	140
Light.....	X, 14, 23, 24, 44, 92, 138, 162, 165
Light radius.....	23
Line of Sight.....	IX, 50, 105, 151
Loading....	VIII, 5-7, 11, 12, 14, 17-20, 24, 32, 41, 43, 44, 49, 54, 60, 66, 73, 136-139, 142
Local.....	13, 69, 72, 74, 81, 82, 85, 90, 125
Lock.....	45, 111, 138, 155, 167, 168
Lockpick.....	45, 49, 138
LoopGroup.....	111, 155
Magic.....	VIII, IX, 14, 15, 28, 35, 37, 41-44, 49-51, 56, 57, 71, 92, 94, 95, 98, 107, 114, 115, 122, 123, 130, 132, 138, 144, 147, 148, 152, 156, 162, 166, 167, 170, 171
Magic effects.....	28, 132
Magicka.....	see Magic
Main.....	33, 52, 76, 131
Markers.....	VIII, IX, X, 48, 49
Master File.....	VIII, 1, 2, 4, 5, 18, 19, 123, 141
Maximum Plug-ins.....	17
MenuMode.....	111, 112, 121, 155
Menus.....	14, 18-29, 93-95, 111, 130, 147, 155, 161, 162
Meshes.....	IX, 2, 3, 7, 10, 35, 39, 40, 56, 176, 177
MessageBox.....	66, 74, 102, 112, 113, 156
MessageBox (function).....	112, 156
Misc item.....	45, 138
MMM Bible.....	see Mod Maker's Bible
Mod-.....	114, 149
-Alarm.....	98, 148
-ArmorBonus.....	98, 148
-AttackBonus.....	98, 148
-Attribute.....	98, 148
-Blindness.....	99, 148
-CastPenalty.....	99, 148
-Chameleon.....	99, 148
-DefendBonus.....	99, 148
-Disposition.....	99, 148
-Fight.....	99, 148
-Flee.....	99, 148
-Flying.....	100, 148
-Hello.....	100, 148

[Table of Contents](#)

-Invisible.....	100, 148
-Paralysis.....	100, 148
-PCCrimeLevel.....	100, 148
-PCVisionBonus.....	100, 148
-Reputation.....	100, 148
-Resist.....	148
-Scale.....	100, 149
-Silence.....	101, 148
-Skill.....	148
-SuperJump.....	109, 148
-SwimSpeed.....	101, 148
-WaterBreathing.....	109, 148
-WaterLevel.....	101, 149
-WaterWalking.....	101, 149
ModCurrentFatigue.....	115, 157
ModCurrentHealth.....	115, 157
ModCurrentMagicka.....	115, 157
ModRegion.....	115, 157
Mod Maker's Bible.....	144
Monetary Result Of Crimes.....	48
Month.....	40, 126, 127, 157, 168
Month (variable).....	127, 157
Morrowind.....	VIII, 1-8, 10-12, 29, 39, 50, 58, 59, 65, 66, 76, 88, 91, 96, 133, 135, 136, 141, 142, 144, 147, 176, 177
Move.....	X, 5, 10, 14, 20, 21, 40, 41, 46, 49, 52-54, 56, 60, 62, 66, 88-92, 96, 97, 104, 109, 115, 119, 139, 147-149, 157, 165
Move (function).....	115, 157
MoveWorld.....	115, 157
Nodes (path grid).....	15, 27, 30, 62, 63
Northmarker.....	IX, 27, 49, 52
NPC.....	IX, 13, 14, 27, 36, 38-51, 57, 58, 62, 66-68, 70, 72, 73, 75, 77, 85, 87, 89, 90, 93 97, 113, 114, 120, 129, 135, 138, 148, 149, 156, 165-167, 169, 170, 171
NPC (object).....	45
OnActivate.....	88, 146
OnDeath.....	87, 116, 157
OnKnockout.....	166, 157
OnMurder.....	116, 157
OnPCAdd.....	127, 157
OnPCDrop.....	127, 157
OnPCEquip.....	127, 128, 157
OnPCHitMe.....	127, 158
OnPCRepair.....	127, 158
OnPCSoulGemUse.....	127, 158
OnRepair.....	128, 158
Paint Shop Pro.....	58
Path Grid.....	15, 27, 30, 54, 62, 89, 138, 146, 163

PayFine.....	116, 158
PayFineThief.....	116, 158
PC Creation.....	see Character Generation
PCClearExpelled.....	116, 158
PCExpell.....	117, 158
PCExpelled.....	117, 158
PCForce1stPerson.....	117, 158
PCForce3rdPerson.....	117, 158
PCGet3rdPerson.....	117, 158
PCJoinFaction.....	117, 158
PCLowerRank.....	117, 158
PCRace.....	113, 128, 145, 156, 159
PCRaiseRank.....	117, 159
PCSkipEquip.....	128, 159
PCVampire.....	129, 159
PCWerewolf.....	129, 159
Persists.....	35, 41, 46, 87, 168
PFS Optimizer.....	141
PKZip.....	9
PlaceAtMe.....	117, 159
PlaceAtPC.....	118, 123, 159
PlaceItem.....	118, 159
PlaceItemCell.....	118, 159
Plane (x, y, z).....	see Axis
PlayBink.....	X, 24, 59, 62
PlayGroup.....	118, 159
PlayLoopSound3D.....	119, 160
PlayLoopSound3DVP.....	119, 160
PlaySound.....	119, 159
PlaySound3D.....	119, 159
PlaySound3DVP.....	119, 159
PlaySoundVP.....	119, 159
Plug-In.....	see mod
Position.....	13, 41, 90, 97, 117-120, 123, 160-162, 168, 169
Position (function).....	119, 160
PositonCell.....	120, 160
Power.....	51, 56, 58, 87, 90, 108
Preferences.....	19, 30, 52, 60
Preview Window.....	X, 52, 54
Probe.....	49, 138
Quests.....	1, 7, 33, 34, 40, 45, 64, 75-78, 90, 105, 11, 112, 125, 126, 131, 135, 139
Quotes.....	12, 79, 91, 112, 114
Race.....	1, 2, 27, 38, 39, 46, 50, 55, 56, 66-68, 70-72, 75, 95, 107, 113, 128-130, 135, 138, 139, 145, 148, 154, 156, 159, 169
RaiseRank.....	117, 120, 159, 160
Ranks.....	46, 63, 64, 67, 69, 71, 106, 113, 114, 117, 120, 153, 156, 158, 159, 160, 169

[Table of Contents](#)

Reactions.....	47, 64, 71, 98, 99, 100, 114, 122, 150
Readme.....	IX, 10, 16, 135, 136, 139, 140, 142
Recompile Scripts.....	31
Redo.....	21, 30, 31
References.....	22, 26, 33, 34, 52-54, 87, 131, 139
Region Map Color.....	26
Region Paint.....	26
Region.....	25, 26, 60, 91, 146, 157
RemoveEffects.....	121, 145, 160
RemoveFromLevCreature.....	120, 160
RemoveFromLevItem.....	120, 160
RemoveSoulGem.....	121, 161
RemoveSpell.....	87, 121, 161
RemoveSpelleffects.....	121, 161
Render Window.....	VIII, X, 16, 20-24, 26, 31, 33-35, 42, 49, 52-54, 60-63
Repair.....	37, 48, 49, 51, 58, 121, 127, 128, 138, 158, 161, 164
RepairedOnMe.....	121, 161
Reputation.....	46, 71, 98, 100, 114, 123, 149, 169
Respawns.....	40, 41, 46, 48, 168
ResetActors.....	13, 161
Resurrect.....	121, 161
Return.....	49
Rotate.....	X, 20, 52, 53, 115, 121, 147, 161
Rotate (function).....	121, 161
RotateWorld.....	121, 161
SameFaction.....	122, 161
Say.....	122, 161
SayDone.....	122, 161
Scale.....	42, 98, 100, 115, 123, 133, 150, 165
Script.....	IX, 1, 2, 11, 14, 15, 17, 22, 28, 29, 31, 35, 36, 38-40, 42-45, 49-51, 53, 63-66, 71-76, 79-130, 135, 138, 145, 146-164
Scripting For Dummies.....	11, 79, 144, 164
ScriptRunning.....	122, 161
Search & Replace.....	22
Semicolon.....	80, 82
SetPos.....	134, 150
Services.....	41, 47, 48
Set-.....	122, 149
-Alarm.....	98, 148
-ArmorBonus.....	98, 148
-AttackBonus.....	98, 148
-Attributes.....	98, 148
-Blindness.....	99, 148
-CastPenalty.....	99, 148
-Chameleon.....	99, 148
-DefendBonus.....	99, 148

-Disposition.....	99, 148
-Fight.....	99, 148
-Flee.....	99, 148
-Flying.....	100, 148
-Hello.....	100, 148
-Invisible.....	100, 148
-Level.....	149
-Paralysis.....	100, 148
-PCCrimeLevel.....	100, 148
-PCVisionBonus.....	100, 148
-Reputation.....	100, 148
-Resist.....	148
-Scale.....	100, 149
-Silence.....	100, 148
-Skills.....	148
-SuperJump.....	109, 148
-SwimSpeed.....	101, 148
-WaterBreathing.....	109, 148
-WaterLevel.....	101, 149
-WaterWalking.....	109, 148
SetAngle.....	123, 161
SetAtStart.....	123, 162
SetDelete.....	123, 162
SetJournalIndex.....	124, 150
SetWerewolfAcrobatics.....	124, 162
ShowMap.....	124, 162
ShowTargets.....	13, 162
ShowVars.....	13, 162
Skills.....	IX, 1, 2, 15, 16, 27, 41, 46-48, 55, 57, 58, 64, 70, 124
Skin.....	IX, 2,3, 38, 39, 176, 177
Skip Initial Cell Loads.....	20
SkipAnim.....	124, 162
Sleep.....	12, 24-26, 60, 106, 112, 126, 153, 164, 171
Sound.....	3, 10, 25, 26, 28, 29, 32, 33, 41, 42, 54, 60, 73, 75, 108, 119, 125, 132-134, 138, 145, 154, 160, 162
Sound Generator.....	29, 32, 75, 133
Spell Effects.....	IX, 1, 2, 28, 35, 43, 45, 49, 50, 100, 101, 108, 120, 121, 132, 145, 160
Spell Effects list.....	145
Spell Points.....	see Magicka
Spellmaking.....	35, 42, 48, 49, 56, 58, 132, 164
StartCombat.....	124, 162
StartScript.....	74, 125, 162
Static (object).....	35, 51, 53, 61, 132, 138
Stats.....	see Attributes
StayOutside.....	74, 75, 129, 162
Stealth.....	41, 57

StopCellTest.....	13, 162
StopCombat.....	124, 162
StopScript.....	81, 125, 162
StopSound.....	119, 125, 162
StreamMusic.....	125, 162
Summary, (plug-in).....	19
Test All Cells.....	27
Test All Land.....	27
Test Dialog.....	see Dialog Conflicts
Test Interior Cells.....	13, 27
Test Models.....	26
Test Non-Teleport Doors.....	27
Test Non-Teleport Interiors.....	27
Test/Fix Path Grids.....	27
TestCells.....	13, 162
TestInteriorCells.....	13, 162
TestModels.....	13, 162
Textures.....	VIII, IX, 7, 10, 15, 35, 39, 59, 61, 141, 142, 176
Tilde.....	5, 118
ToggleAI.....	13, 162
ToggleBorders.....	14, 162
ToggleCollision.....	14, 162
ToggleCollisionBoxes.....	14, 163
ToggleCollisionGrid.....	14, 163
ToggleDebugText.....	14, 163
ToggleDialogueStats.....	14, 163
ToggleFogOfWar.....	14, 163
ToggleFullHelp.....	14, 163
ToggleGodMode.....	14, 163
ToggleGrid.....	14, 163
ToggleKillStats.....	14, 163
ToggleLights.....	14, 163
ToggleLoadFade.....	14, 163
ToggleMagicStats.....	14, 163
ToggleMenus.....	14, 163
TogglePathGrid.....	15, 163
ToggleScriptOutput.....	15, 163
ToggleScripts.....	15, 163
ToggleSky.....	15, 163
ToggleStats.....	15, 163
ToggleVanityMode.....	15, 163
ToggleWater.....	15, 163
ToggleWireframe.....	15, 163
ToggleWorld.....	15, 163
Top Down View.....	see Isometric View
Total Conversion.....	135

[Table of Contents](#)

Towns.....	12, 148
Travel.....	X, 5, 23, 48, 49, 89, 93, 103, 106, 107, 131, 146, 151, 153, 164
Tribunal.....	VIII, IX, 2, 6-8, 11, 24, 25, 29, 73, 96, 100, 135, 142, 176
TurnMoonRed.....	125, 163
TurnMoonWhite.....	125, 164
Tutorials.....	5, 30, 59, 77, 78, 141, 144
Undo.....	21, 30, 61, 65, 125, 164
UndoWerewolf.....	125, 164
Unlock.....	126, 164
Updating A Plug-in.....	6
Use Angle Snap.....	20, 30, 53
Use Grid Snap.....	30
UsedOnMe.....	126, 164
Utilities.....	3, 7-11, 16, 18, 49, 136, 137, 141-143, 176
Variables.....	29, 81, 97-110, 112-115, 126, 149, 164
Variables, Float.....	29, 54, 72, 81, 82, 90, 113, 126-129, 145, 147, 156-159, 162, 164
Variables, Global.....	7, 13, 22, 56, 57, 69, 88, 100, 102, 103, 116
Variables, Local.....	13, 69, 72, 74, 81, 82, 85, 162
Variables, Long.....	29, 81, 113
Variables, Short.....	29, 75, 81, 99-111, 113, 126, 147
Vertices.....	60, 61
WakeUpPC.....	126, 164
Warning Messages.....	see Error
Water Level.....	24, 101, 150
Weapon.....	1, 7, 40, 42, 47-49, 51, 87, 90, 92, 94, 98, 99, 109, 110, 114, 122, 132, 133, 138, 149, 155, 166, 168-171
Weather.....	23, 25, 26, 31, 60, 71, 91, 103, 115, 146, 151, 157
WinAce.....	9, 175
WinRar.....	9, 175
WinZip.....	9, 175
Year.....	129, 164
YesToAll Button.....	4
ZipGenius.....	9, 175