

IceCC: User Manual

Version: 1.2b (Updated: 2002/06/08)

by Jeff Pang <jp@magnus99.dhs.org>website: <http://magnus99.dhs.org>**Table of Contents:**

I.	Introduction	2
II.	Installation	4
III.	Iscript.bin Refresher	5
IV.	Graphical User Interface	7
V.	Command Line Interface	9
VI.	Script Files	13
VII.	Example	17
VIII.	Appendix A: Animation Types	20
IX.	Appendix B: Instruction List	23
X.	Appendix C: Script File Specification	29
XI.	Appendix D: Other Useful Programs	31
XII.	Appendix E: Program Configuration	32
XIII.	Appendix F: DAT Entry Lists	33

Introduction

What is IceCC?

IceCC is a pair of programs, `icedc` and `icecc`, that decompile and compile the Starcraft `iscript.bin` (image animation script) file. It is used to create your own animations for Starcraft in-game graphics such as unit sprites.

Should I be using IceCC?

If you are new to Starcraft customization and 'mod making' then IceCC is probably not the best place to start. This manual assumes the reader is familiar with tools such as Stardraft and/or MPQ 2000 (or some other mpq editing tool). A good place to find other Starcraft editing is Camelot Systems (<http://www.camsys.org>); I've also written a beginner's tutorial on Starcraft editing called the 'Starcraft Editing Bible' which you can download (in HTML format) on my website (<http://magnus99.dhs.org/downloads/>). It is the zip file called 'sceb-* .zip' where * is the last date it was updated.

What's new in version 1.1?

No new functionality was added. But, I spent a day and a half learning Java Swing and wrote a graphical user interface for IceCC called IceCCUI. Windows users who like point and click rejoice! :) See the new section called Graphical User Interface for more information.

How is IceCC different than ICE?

ICE, the iscript editing tool by KramerBoy (Camelot Systems), is a graphical program and edits the `iscript.bin` file directly. It has several advantages over IceCC; for example, it allows you to preview GRP images (Starcraft bitmap collections) and WAV sounds associated with some animations. It also has a graphical user interface and hence may be more familiar to most Windows users. However, it does have some pitfalls; some of its instruction information is wrong and it can be hard to navigate when a user wants to add new animations. IceCC is meant to rectify these flaws in as simple a manner as possible. IceCC allows you to 'extract' animations from the `iscript.bin` binary file to a human readable text file for editing. Animation and instruction information is up to date and complete so it functions without errors. After editing the text scripts, IceCC can 'compile' them back into an `iscript.bin` file seamlessly. But most importantly, IceCC runs on Mac OS X as well. :)

How can I help?

The most important thing that is still unfinished right now is the list of iscript instructions. Most of them have been identified already and all of their formats have been identified completely, but I'm still not sure what all of them do. If you could experiment with some of the unknown instructions (and seeing how they are used by animations in the default `iscript.bin` file) and try to figure out what they do, that would help others who might want to use them. See Appendix B for a list of instructions.

Legal Stuff.

IceCC. The Starcraft Iscript Compiler/Decompiler
Copyright (C) 2000-2002 Jeff Pang <jp@magnus99.dhs.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Installation

If you're reading this manual, there is a good chance that you've already got IceCC installed.

Windows:

IceCC should have been distributed to you in a zip file named something like 'icecc-**-win32.zip*' where * is the version number. Simply unzip all the files to C:\. This will create a directory called C:\IceCC where the program files will be installed. If you wish to install IceCC elsewhere, you can do so, but you will have to edit the icecc.ini file in the IceCC directory to reflect where you performed the installation (change where it says INSTALLDIR=C:\IceCC to INSTALLDIR=*[install path here]*).

To run the graphical version of IceCC (IceCCUI), you will need a Java Runtime Environment version 1.2 or greater. You can get it from Sun here: <http://java.sun.com/j2se/1.3/jre/download-windows.html>. Make sure that the 'java' family of programs was added to your PATH (which is automatically done by the Sun installer, but may not have worked if your computer crashed during installation or something weird). This is almost surely the problem if you are having problems running IceCCUI.

To run IceCCUI (the graphical version of IceCC), just double click on the IceCCUI.BAT file in the installation directory. If you have file extensions turned off, then it will just be called IceCCUI and have a little yellow gear icon inside a little window.

To run icecc or iccdc, you will first have to open an MS-DOS prompt and change your working directory to C:\IceCC (or where you performed the installation) by using the 'cd' command. If you wish to be able to run icecc and iccdc from any directory, you will have to open a file called C:\AUTOEXEC.BAT in a text editor and find the line that starts with 'SET PATH=...' where ... looks like a list of directory names separated by semi-colons (;). At the end of that line, add ';C:\IceCC' (a semi-colon followed by the installation directory). Then restart Windows and you should be all set.

Mac OS X:

IceCC should have been distributed to you in a tar.gz file named something like icecc-**-macosx.tar.gz* where * is the version number. Double click this to expand it and drag the resulting directory (Ice CC) to your Hard Drive somewhere.

To run the program, double click on the IceCC icon in the folder (Dark Templar).

To run the command line versions, you can open up a terminal, enter the Ice CC directory and call ./icecc and ./iccdc for the compiler and decompiler respectively. If you know a little about UNIX, then you can add the directory to your PATH, etc. since they are regular UNIX binaries.

Iscrip.t.bin Refresher

To use IceCC, or any iscript.bin program, you will need to know a little bit about the iscript.bin file. It is the file that controls the animation of graphics in Starcraft, or how the units in the game 'move.' For example, it determines how a typical marine would walk, how a battle cruiser flies, and the way a hydralisk flexes its gills when it is idle.

There is a unique distinction between a 'unit' in starcraft and a 'graphic' or 'sprite.' A unit is the physical 'thing' that is selectable and controllable in the game, while a sprite is just how it looks on the screen. Any unit can practically change its graphic at will (with some editing of the DAT files) without affecting its other properties. For a more detailed explanation, you should see my 'Starcraft Editing Bible.'

Let's fast forward a head a bit and talk about DAT files. DAT files are the main configuration files which control various features about the way Starcraft behave. There are 4 DAT files which we are concerned with: units.dat, flingy.dat, sprites.dat, and images.dat. Units.dat controls unit properties and includes a variable which determines which graphic it uses. This variable is really just a 'pointer' to a flingy.dat entry, which is a list of movable sprites (units, projectiles, etc.). Flingy.dat contains a pointer to a sprites.dat entry, which is a larger list of sprites (including all the unit and projectile graphics in addition to things like doodads and terrain graphics). Sprites.dat in turn has a pointer to an images.dat entry, which is the most 'raw' graphic description of them all. Images.dat contains almost 1000 different graphic descriptions. There are so many because these include things like the 'engine glow' that the afterburners of a Valkyrie make or the ruins that are left on the ground when a Terran building is razed. Every images.dat entry has a pointer to a GRP file; this file contains the actual set of bitmap images or 'frames' that the graphic will use. The images.dat entry also contains a pointer to an 'Iscrip ID', which, as you might have guessed, determines which set of animations it uses. This is where IceCC takes over; it allows you to edit, or even generate completely new animation sets. For a in depth explanation see my 'Starcraft Editing Bible.'

You don't really have to know about all the connections between the DAT files to use IceCC, because it will allow you to extract animation sets from the iscript.bin based on units.dat entry (by following the pointers mentioned above), but they are good things to keep in mind just so that you understand how it all works. Each iscript animation set is composed of several 'animation scripts.' I will refer to the animation set itself has an iscript 'header' because it is like a header which determines what animations are actually used. A header looks something like this:

```
IsId      10      # this is the iscript id of this animation set; it is
              # referenced by images.dat, each set has a unique id
Type      0      # this is the type of set; there are 28 different
              # types, each with a different number of animations
Init      BloodInit # this is one of the animations in this
              # set.
Death     BloodDeath # this is another animation in the set
```

This example shows the animation set which has an iscript ID of 10. It is of type 0 which means it has two animations, Init (the initialization animation) and Death (the death animation). See Appendix A for a info on all the set types. The 'label' after each of these animation tags is the name of the actual animation scripts that will be used. These scripts are a series of instructions which determine what the animation does (i.e., what order to play the image frames, when to turn the graphic, when to play sounds, etc.). An animation script might look like this:

```
BloodInit:
    playfram    1
    wait        2
    playfram    2
    wait        2
    playfram    3
    wait        2
    playfram    4
    wait        2
    goto        BloodInit
```

This simple script, which is the script pointed to by the Init animation of the above iscript header, tells Starcraft to play the sprite's frames 1 through 4, waiting 2 ticks between each (this gives you a sense of fluid motion). Then the script says 'goto BloodInit' which just means, loop back to the top (loops are very common in the iscript, because, unless you want your graphic to disappear right away, loops allow for the animation to play continuously without writing a long script).

And that's all the iscript is: a collection of headers (sets of animations) and a listing of scripts. Of course, most headers have more than two animation scripts. For 'regular' unit graphics, for example, like the Marine's header, there will be scripts for the initial animation, the death animation, the walking animation, the attacking animation, etc. These will be explained further in depth later (and fully in Appendix A). Again, for further information, you can see the 'Starcraft Editing Bible.'

Graphical User Interface

Version 1.1 added a new graphical user interface written in Java Swing. You will need a Java Runtime Environment (version 1.2 or greater) to run it (see the Installation section). The graphical interface mimics the command line interface, so if you're already familiar with the later then you can probably figure out the former for yourself.

To get started, double click the `IceCCUI.BAT` file in the IceCC installation directory. (You can create a shortcut to this if you want) This should bring up IceCCUI ("Iscrip Code Compiler User Interface"). There are two tabs, one for decompiling iscript.bin files and one for compiling script files back into binary iscript.bin files starcraft can use.

Decompiler Tab

The decompiler tab has 4 main menus and several options at the bottom. Use the menus to select which iscript.bin entries you want to decompile to a script file. Each list refers to a DAT file that you can reference entries from. (e.g., if you select "0 Scourge" out of the Images list then you will get the images.dat scourge entry's iscript animation set; if you select "0 TerranMarine" from the Units list then you will get the animation set that is associated with the Terran Marine) Why all these lists? Because not all the animation sets can be traced back to a units.dat entry, so the Units list is too short, but in many circumstances, you do not know the exact images.dat entry which corresponds to the iscript animation set you want, but you know the Units entry, or flingy entry, so these lists are convenient. Click on an entry to select it/deselect it. You can use ranges by holding down shift.

Below the lists is a text field where you can enter iscript ID numbers of animation sets explicitly. E.G., if you know you want exactly the animation sets with iscript ID 0, 10, and 54, then you'd enter "0,10,54" in the text field. Entries are separated by spaces, commas, or semi colons. If you do not select any iscript IDs explicitly *and* you don't choose any items from any of the lists, it is assumed you want to decompile *all* the entries in the iscript file.

The option 'Use default iscript.bin' will make the decompiler decompile entries from the default iscript.bin that is in Starcraft Broodwar 1.07. If you uncheck the option, then you can select your own iscript.bin file to decompile in the 'Open' field. The 'Separate Headers' option makes sure that no two animation sets share any animation routines. This will make more sense later when you read the script file examples.

The 'Save To' field allows you to choose where you want to save the generated script file to. If you leave it blank, it will be saved to 'iscript.txt' in the current directory. The 'Open in Editor' button allows you to quickly open this file in a text editor (see Preferences to choose your favorite text editor).

When you've set up all the options, then you can click 'Decompile' and the decompiler will create your script file. If nothing happens when you click it, that means it worked. If you get error dialogs, that means it didn't. :) Read the error messages and see if you can fix whatever the problems are.

Compiler Tab

The compiler tab allows you to compile your script files back into a usable iscript.bin file. Use the 'Add' and 'Remove' buttons to manage the source script files you want to compile. The files closer to the bottom of the list have HIGHER precedence than the ones near the top; this means that if two of them have the same iscript ID in them (e.g., they conflict), then the one in the file closer to the bottom will be used and the one in the file closer to the top thrown out. You can adjust the positions with the 'Up' and 'Down' buttons. You can also use the 'Open' button to quickly open a selected file in a text editor for viewing.

The 'Merge with default iscript.bin' option means that all your scripts will be merged with the full iscript.bin. This is probably what you want, since it will just add your modified entry back into the original iscript

file. The 'Display all warnings' option will make the compiler display all the warnings and errors it encounters trying to compile your scripts. If unchecked, only the first 10 errors/warnings will be displayed.

The 'Save to' field allows you to choose the name of the iscript.bin file that will result from the compilation. If you leave it blank, it will be 'iscript.bin' in the current directory.

Click compile when you have everything set up. Again, nothing happening means it worked, otherwise the compiler will give you an error dialog. If the dialog has a bunch of syntax errors in the message, you can click on an error and choose to open that file in a text editor for quick viewing.

File->Preferences

The preferences dialog has a few settings you can change. The 'icecc Executable' and 'icedc Executable' options determine where IceCCUI will find the command line IceCC programs. The 'Config File' and 'Config Dir' options set where to find the configuration files IceCC needs. See the last section of this manual for more info. The 'Text Editor' option allows you to specify which text editor program (or any program really) you want to launch when 'Open' buttons are pressed, etc. This is something like 'notepad' or 'wordpad' or the full path to a text editor executable like 'C:\Program Files\MS Office\MSWord.EXE' or 'C:\Program Files\UltraEdit\uedit.exe'. Find out where your favorite text editor is installed.

These preferences are written to a text file called 'iceccui.ini' in the current directory. If you are getting error dialogs about preferences, then you might want to check that out.

This primer on the GUI is pretty short, and it will probably benefit you to read the Commandline section if you don't know what everything does yet, even if you're never going to use it. The GUI just reads in your options and runs the command line programs anyway (They're the guts, IceCCUI is just a pretty face on top :).

So now you've saved some script files for editing. Now what? See the 'Script Files' section for the details.

Command Line Interface

Icecc and icedc are command line programs. Instead of double-clicking on an icon to pop up a window, you will run them by entering commands in an MS-DOS prompt (which is called a 'shell' in CLI land). Here's how to get started:

1. Open an MS-DOS prompt. Normally, you can find this in the **Start Menu->Programs->Accessories** submenu.
2. Change the working directory to C:\IceCC or where you chose to install IceCC by typing 'cd C:\IceCC'. You don't have to do this if you set up your AUTOEXEC.BAT file to allow you to run IceCC in any directory. (See Installation)
3. Now you're set to enter commands.

If your command-line skills are a bit rusty, a good tutorial may be helpful. You can find one at the FreeDOS project here <http://www.freedos.org/fd-doc/mini/tutor/menu.html>.

Mac OS X users: If you are familiar with the command line, then I don't think I should have to explain the differences between the DOS instructions and the UNIX ones (hint: instead of the **C:>** prompt, you have a nicer **tcsh** one :). But do remember that Mac OS X file paths are not bastardized like Windows and use forward slashes "/" and not back slashes "\". There are many Terminal tutorials for OS X online, so they would be the best resources to consult to learn more about the UNIX layer in Mac OS X.

IceCC's command-line options are consistent with UNIX standards. Naturally, I don't expect you to be familiar with this standard, but if you are, you can skip this part. As an example, I'll use icedc, the iscript decompiler. To run the decompiler, at the prompt you would type:

```
C:\IceCC> icedc [options] <inputfile>
```

Where the *<inputfile>* is the name of the iscript file you wish to decompile and *[options]* is a list of optional "switches" you wish to give to the program. ("**C:\IceCC>**" is the prompt of course, so you won't type that) Let's say you have the iscript.bin file in your "My Documents" directory. Do decompile it you would type:

```
C:\IceCC> icedc "C:\My Documents\iscript.bin"
```

In this case, you must put the file path in quotes because the "My Documents" directory has a space in its name. This command would decompile the iscript.bin file to the default output-file "iscript.txt" in the working directory. So you could open that file up (such as with the command "notepad C:\IceCC\iscript.txt"; though you probably want to use a better text editor than Notepad) and edit it.

This is really all you need, but generally you will never use the above command line. This is because, it will decompile the *entire* iscript.bin to iscript.txt, creating a text file with over 20,000 lines. If you want to explore the original iscript animation sets, this may be useful, but in general, you will only want to deal with a few sets at a time. To facilitate this, icedc allows you to extract animation sets by their corresponding iscript ID (which you can find out from images.dat). For example, let's say I only want to extract the sets with ID 0 and 2:

```
C:\IceCC> icedc -i 0,2 "C:\My Documents\iscript.bin"
```

The "-i" is known as an option or a switch, all of which begin with a "-" and consist of a single letter. In particular, this option followed by a comma-separated list of numbers tells icedc to only extract the sets from the iscript.bin file which have the Ids 0 and 2 (these are the Scourge and Scourge Death animation sets respectively). What if you don't know the iscript ID of the animation set you want to extract? Icedc will allow you to extract them via images.dat entry number also:

```
C:\IceCC> icedc -m 0 "C:\My Documents\iscript.bin"
```

This will extract the iscript ID associated with images.dat entry number 0 (which just happens to be the scourge also. If you don't know the images.dat number, you can extract by sprites.dat entry number, or flingy.dat entry number, or even units.dat entry number. Here are the corresponding switches, each of which should be followed a comma-separated list of entry numbers you wish to extract:

```
-i    iscript ID numbers
-m    images.dat entry numbers
-p    sprites.dat entry numbers
-f    flingy.dat entry numbers
-u    units.dat entry numbers
```

You can also use any combination of options in conjunction:

```
C:\IceCC> icedc -i 0,2 -m 10 -u 0 "C:\My Documents\iscript.bin"
```

This extracts iscript IDs 0 and 2 along with the ones associated with images.dat entry number 10 and units.dat entry number 0.

One thing to remember is that it is very possible that two images.dat entries share the same iscript ID. This means changing that animation set for one of the entries will change it for all of them. The decompiled iscript.txt file will have comments telling you which image.dat entries use its animation sets. If you want to two images.dat entries which normally use the same iscript ID use separate ones, then you will have to also edit the images.dat file that will be a part of your custom. (And then you can use IceCC to create a unique iscript ID and a new set of animations for your new image)

Of course, it is cumbersome to have to specify a input file all the time, since normally you will always extract your animation sets from the default iscript.bin file that is already in Starcraft. The '-d' option tells icedc to use the default iscript.bin file. This does the same as the above using the default file:

```
C:\IceCC> icedc -d -i 0,2 -m 10 -u 0
```

The order of options doesn't matter, and the interface is pretty flexible. For example, you can combine two or more options behind the same '-' if they do not expect something after it:

```
C:\IceCC> icedc -di 0,2 -m 10 -u 0
```

This combines the d and the i option. We can't put the m option together with the pair because i expects a list of numbers after it.

Here is a full listing of the command line options:

```
-h    display a help message summarizing all the options.
-v    display the version of the program
-d    decompile the default iscript.bin file
-s    separate the headers for animation sets (explained later)

-o <outputfile>    decompile to <outputfile> instead of the default iscript.txt
-c <configfile>    use configuration file <configfile> (explained later)
-r <configdir>     use the configuration directory <configdir> (explained later)
-i <iscriptidlist> only decompile iscript IDs in the list
-m <imageslist>   only decompile those associated with the images.dat list
-s <sprites.dat>   only decompile those associated with the sprites.dat list
```

<code>-f <flingylist></code>	only decompile those associated with the <u>f</u> lingy.dat list
<code>-u <unitslist></code>	only decompile those associated with the <u>u</u> nits.dat list

Icecc, the iscript compiler works a similar way:

```
C:\IceCC> icecc [options] <inputfile(s)>
```

This time, the program can take multiple input files, each of which is either a iscript.bin file or a text file that the decompiler generated and you edited (or you wrote from scratch). The Icecc program will merge all of these files and output a single iscript.bin which consists of all of the header sets which were in each of the files. For example, let's say you extracted the header set with iscript ID 0 (the scourge animations) into iscript.txt, edited them, and now you want to merge it back into your iscript.bin file. Here is what you can do:

```
C:\IceCC> icecc "C:\My Documents\iscript.bin" iscript.txt
```

This will merge your iscript.txt file with the original iscript.bin file and output a file with the default name "iscript.bin" in the current directory. If you have more than one set of animations decompiled into separate files, you can merge those in too. You can even merge multiple binary iscript.bin files if you happen to find two with different sets of animations (this may happen if custom iscript.bin files start to be made by others):

```
C:\IceCC> icecc iscript1.bin iscript.txt iscript2.bin iscript2.txt
```

Any file with a "bin" extension is assumed to be a binary iscript.bin and all other files are assumed to be text scripts. Normally you will just want to merge your changes with the original iscript.bin, so the '-m' merge option facilitates this:

```
C:\IceCC> icecc -mo newis.bin iscript.txt iscript2.txt
```

This example also makes use of the '-o' option which tells the compiler to output the compiled file with the name 'newis.bin' instead of the default 'iscript.bin'. Note that there may be conflicting Iscript IDs in two files. E.G., when compiling your animation header with ID 0 into the original iscript, you are overwriting the animation header with ID 0 in the original iscript.bin. This is OK, and the compiler will deal with it. This is the rule: the files closest to the *right* have precedence; so in the example above, if iscript.txt and iscript2.txt each had a header with the same iscript ID, the one in iscript2.txt would win and actually get used. The other one is thrown out. When using the '-m' merge option, the original iscript.bin has lowest precedence (naturally, since you want to use your scripts instead of the originals).

Besides its use as a compiler, icecc also serves as a debugger. While the decompiler will always output text scripts that will compile without any problems, when you edit them, you will undoubtedly make mistakes, whether it be misspelling an instruction name or forgetting to write in a animation label. When you try to compile a text script with syntax errors in it, icecc will try to tell you where they are, what they are, and maybe what you can do to fix them. Sometimes, they will only be 'warnings' and icecc will attempt to recover for you and will not abort the compilation (though warnings generally suggest you still made a mistake, it just wasn't a "fatal" one). In any case, a syntax error will generally mean that the compiler will abort and will not generate any output file (a specific error message at the end will tell you so). These debugging messages will be covered in further detail in the next section (Script Files).

Here is a full command-line overview of icecc:

<code>-h</code>	<u>h</u> elp message
<code>-v</code>	program <u>v</u> ersion
<code>-m</code>	<u>m</u> erge with the original iscript.bin
<code>-w</code>	display all syntax errors and <u>w</u> arnings (usually only the first 10 are shown)

<code>-o <outputfile></code>	compile to <outputfile> instead of the default iscript.bin
<code>-c <configfile></code>	use config file <configfile> (explained later)
<code>-r <configdir></code>	use config directory <configdir> (explained later)

You can always get a summary of all the commandline options to either icedc or icecc by running the programs with no arguments:

```
C:\IceCC> icedc
C:\IceCC> icecc
```

Script Files

Section III already introduced the general look of the iscript decompiled scripts, but here we'll take a in depth look. Each animation set consists of an iscript header that looks like the following:

```
# ----- #
# This header is used by images.dat entries:
# 000 Scourge (zerg\avenger.grp)
.headerstart
IsId          0
Type          12
Init          ScourgeInit
Death         ScourgeDeath
GndAttkInit   [NONE]
AirAttkInit   ScourgeAirAttkInit
SpAbility1    [NONE]
GndAttkRpt    [NONE]
AirAttkRpt    ScourgeAirAttkInit
SpAbility2    [NONE]
GndAttkToIdle [NONE]
AirAttkToIdle ScourgeAirAttkToIdle
SpAbility3    [NONE]
Walking       ScourgeWalking
Other         ScourgeAirAttkToIdle
BurrowInit    [NONE]
.headerend
# ----- #
```

The first thing to note is the lines that begin with a hash or pound-sign character (#). The hash marks the beginning of a "comment" and the rest of that line is ignored by the compiler. So that means you can use comments to make notes to yourself. Secondly, note the ".headerstart" and ".headerend" tags. These two tags are special identifiers that mark off the beginning and the end of a header entry, respectively. Everything in between the two is considered to be part of that header (like all text in the script, they are case-sensitive). Inside the header, there are the two special tags, IsId and Type, which determine the unique iscript.bin ID and the type of animation set, respectively. Type number 12 has the 14 animations listed: Init, Death, GndAttkInit, AirAttkInit, SpAbility1, GndAttkRpt, etc. Each of these animation variables points to a label name in the code, which tells Starcraft where they start. For example, The AirAttkInit animation starts at the label ScourgeAirAttkInit. While the Type always determines exactly how many animations a set has, there is no need for every set to use *all* of them. For example, the scourge doesn't need the SpAbility1 animation which is used for spell casting, since it's unit doesn't use any spells. Therefore, we can give it the special label [NONE]. This is the most common reasons why when you switch graphics between units in the units.dat file (with a tool like Arsenal III), starcraft will sometimes crash. E.G., if we were to give, say, the queen the scourge graphic, the game would crash when we try to cast a spell, because, as you see above, the scourge graphic's animation set doesn't have an spell casting animation. We can fix this easily by changing [NONE] to an actual label in the code.

Between header entries, there is the actual animation scripts that determine how the animations are played. For example, here is the scourge Init animation:

```
ScourgeInit:
    imgul09          1 0 42 # ScourgeShad (zerg\avenger.grp)
    playfram         0x00 # frame set 0
    waitrand         1 5
ScourgeAirAttkToIdle:
    playfram         0x00 # frame set 0
    shvertpos        0
    wait             3
    playfram         0x11 # frame set 1
    wait             3
    playfram         0x22 # frame set 2
```

```

shvertpos      1
wait           3
playfram       0x33    # frame set 3
shvertpos      2
wait           3
playfram       0x44    # frame set 4
shvertpos      1
wait           3
goto           ScourgeAirAttkToIdle

```

Each script begins with a label, and then proceeds with lines of instructions. You will notice that in the middle of this Init animation script, the AirAttkInit script also starts with the label ScourgeAirAttkInit. This demonstrates a key feature of the iscript: different animations can share code. In fact, different *animation sets* can share code (you can have the firebat and marine have the same GndAttkInit script for example). This is very useful if you don't want to have to rewrite a lot of stuff for many different animation sets. However, it can also become hairy if you have too many headers jumping around everywhere. This is the nature of coding. :)

The other important thing to note is that this script "ends" with a goto instruction that basically makes it loop (though not all the way back to the beginning). Every chunk of code you write should always end in a loop or a "end" instruction, and only the latter if you actually want the animation to stop completely and have the graphic disappear from the screen. The most common way this is done is by just looping a script back to where it started, or somewhere in the middle of it. Here is another example:

```

ScourgeWalking:
    shvertpos      0
local01:
    playfram       0x00    # frame set 0
    wait           2
    playfram       0x11    # frame set 1
    wait           2
    playfram       0x22    # frame set 2
    wait           2
    playfram       0x33    # frame set 3
    wait           2
    playfram       0x44    # frame set 4
    wait           2
    goto           local01

```

This scourge walking (or rather, flying) animation loops back to its second instruction. Loops can be created by simply creating a label before an instruction and then using a goto to jump back to that label. Label names can contain any characters except for spaces and tabs, and always end in a colon when you declare it. You can "goto" anywhere in the same file, so long as your label names in your goto and before the instruction match. The same goes for the label name used in the header.

Two other little tidbits should be of note here. First, the numbers for each "playfram" instruction are in hexadecimal (base 16 instead of base 10). If you don't like this, then you can use regular based 10 numbers by excluding the '0x'. However, this is useful for the iscript animations which are used by "turning" images (this is a variable set in images.dat). These images can face multiple directions, and they will look different depending on whether they are facing north, west, east, or south. In fact, each turning image has 17 different bitmap images representing each angle it can face (there are actually 32 angles it can face, but the latter 15 are just mirror images of the first 15). You will notice this if you look through GRP images. When playing frames for these turning graphics, the iscript uses the same animation no matter which direction the animation is facing; however, Starcraft also adds a number from 1 to 16 to this number (depending on which angle it is at) to make sure the correct bitmap image corresponding to its direction it is facing is displayed. What should you get out of all this? Mainly this: since each "set" of bitmaps is 17 frames, then the 0th set always begins with frame 0x00 (0), the 1st set begins with 0x11 (17),

the 2nd set begins with 0x22 (34), the 3rd set begins with 0x33 (51), the 4th set begins with 0x44 (68), all the way up to the 15th set at 0xFF (255). Since hex notation makes reading "frame sets" instead of individual frames easier to read, hex numbers are used instead. If you really don't like hex, then just don't use the notation yourself. If you really like hex, you will be happy to know that you can use hexadecimal numbers instead of decimal in any instruction or place where a number would go (just prefix it with '0x'). You can also use octal notation (base 8) by prefixing your number with a '0' (zero). Dunno why you'd want to do that, but just in case you accidentally write 010 and wonder why its value is 8 and not 10. :)

That's basically it. Scripts consist of a header and a series of instructions grouped together in scripts. To learn more about the nuances of all the different instructions that are available to use in your scripts, see Appendix B.

Inevitably, when you write scripts, you will make errors: typos, stuff you forgot to write down, or simply things you didn't think about. Icecc, the compiler, will try to tell you where these syntax errors are in your script files, and how you can go about fixing them. Sample output by icecc trying to compile a poorly written script might look like this:

```
scourage.txt:19: error: unknown animation name 'SpAility2'
scourage.txt:45: error: unknown instruction opcode 'shvertos'
scourage.txt:63: error: not enough arguments to instruction 'wait'
scourage.txt:64: error: instruction 'goto' not connected to a label;
discarding instructions until next label
scourage.txt:70: error: too many arguments to instruction 'wait'
scourage.txt:79: error: final instruction does not terminate
scourage.txt:60: error: can't find label name 'local00'
scourage.txt:55: warning: label 'ScourageAirAttkInit' is not referred to by
any headers or instructions
scourage.txt:55: warning: label 'ScourageAirAttkInit' is unreachable
scourage.txt:28: warning: label 'ScourageInit' is not referred to by any
headers or instructions
icecc: error: aborted due to syntax errors in scourge.txt
```

It looks a bit terse (and dirty) at first, but the formatting of the error messages is consistent and are short and easy to read once you get used to it. Each error message is formatted like the following:

```
<filename>:<line number>: <error or warning>: <error message>
```

The <filename> is the file that it was trying to compile. The line number is where the error occurred in the file (or the compiler's "best guess", it may be "around" that area). Let's go through each of these errors one by one. The first error on line 19 says that the compiler doesn't know an animation by the name of 'SpAility2'. Looking in our file, we see that is in a header and that we probably meant to write 'SpAbility2' (missed the 'b', whoops :). At line 45, the compiler doesn't understand the instruction 'shvertos'. Again, this is a small type we can fix by writing 'shvertpos' instead. On line 63, we forgot to give an argument to wait (wait needs to know how many clock ticks to wait, I didn't write anything after it). On line 64, we have a strange error. Looking at the actual file, we see this:

```
...
59:      attack25          2
60:      goto              local00
61:
62: local00:
63:      wait
64:      goto              local00
65:
66: ScourageWalking:
57:      shvertpos         0
```

...

Line numbers shown for clarity. The error tells us that the 'goto' instruction on line 64 is not connected to any label, but here we see clearly that it can be reached by starting at the label 'local00:'. However, looking back at our previous error, we see that we messed up the wait instruction on line 63; this caused the label 'local00:' not to register with the compiler (since the instruction it was pointing to was erroneous). Hence we get this extraneous error message. Sometimes it is useful to fix the first few errors that make sense to you and try to compile again. Some of the later error messages may only be a result of former errors and will disappear once you fixed the earlier ones.

The next error on line 70 shows us that I accidentally wrote 'wait 2 4' when I meant to write just 'wait 2'. On line 79, I have the following:

```
...
75:  wait                2
76:  playfram           0x33  # frame set 3
77:  wait                2
78:  playfram           0x44  # frame set 4
79:  wait                3
```

Which is followed by the end of the file. Thinking for a moment, I realize that this script just trails off the end without a terminating 'end' instruction or a 'goto' to loop back to a label. I meant to terminate this off with a 'goto local01' which is a label just above to loop it. Next on line 60, it says that the label local00 was not found. Looking back at the first example, this means I used local00 in a goto instruction, but don't actually have a 'local00:' label pointing to any instruction. Strange, I see that I wrote local00: right above that; but, this again is the result of that bogus 'wait' instruction. Since the instruction the label was pointing to was erroneous, it never registered and hence, the compiler doesn't know what it is. Again, this is one of those extraneous errors that would go away after fixing the first few easy ones. Next we have a pair of warnings on line 55; that the label 'ScourgeAirAttkInit' is not used by any goto instructions or by any header animations and, hence, that it is not reachable (meaning there is no way we can get to that label). This is only a warning, since the compiler will just discard the label and hope that everything turns out OK. If we look back from the way beginning, we see that this error occurred because we misspelled 'SpAility2' in the header, and thus that animation's label pointer (which was this label that the compiler says is unreachable) was discarded. Finally we have 'ScourgeInit' which the compiler says is not used by any headers to goto instructions. This error, believe it or not, is a result of the 'SpAility' misspelling also. Because we fudged the animation name in the header, the entire header did not register with the compiler, and hence while the label 'ScourgeInit' is referenced in our header, it did not register, since our header did not register.

The moral of the story? A lots of error messages doesn't mean you have lots of errors. Fix the ones you understand and usually the rest will go away. The compiler just tries to be extra helpful just in case you can understand all of them. :)

For more information, see the Appendices.

Example

This section will go through a quick example. One day, we looked at the SCV while it wasn't doing anything and we thought, "hey it's not even moving." It would be much more exciting if the SCV did something while it was idle; even the ghost likes to cock his gun every once in a while. So we decide to do something about it.

First we need to extract the SCV's animation's from the iscript.bin. I don't know the SCV's iscript ID, and I don't feel like looking through all 1000 images.dat entries looking for it, so I fire up Arsenal III and look for the SCV units.dat entry (the entry number is displayed in the lower right hand corner). It is number 7.

In IceCCUI, I select the **Decompiler Tab**. Then I select "7 TerranSCV" in the **Units** list. Next, I type in `scv.txt` in the **Save to:** field so that that's where the script will be saved to. Finally, I click **Decompile**.

Now, I open up `scv.txt` in my favorite text editor. Notepad in Windows and TextEdit in Mac OS X are good enough, but you can easily do better with something like UBB Edit (Windows) or BBEdit (Mac).

I see the following:

```
# ----- #
# This is a decompile of the iscript.bin file './data/scripts/iscript.bin'
# created on: Sun Jan 14 19:30:03 2001
# ----- #

# ----- #
# This header is used by images.dat entries:
# 247 SCV (terran\SCV.grp)
.headerstart
IsId                84
Type                15
Init                SCVInit
Death              SCVDeath
GndAttkInit        SCVGndAttkInit
AirAttkInit        [NONE]
SpAbility1         [NONE]
GndAttkRpt         SCVGndAttkInit
AirAttkRpt         [NONE]
SpAbility2         [NONE]
GndAttkToIdle      SCVGndAttkToIdle
AirAttkToIdle      [NONE]
SpAbility3         [NONE]
Walking            SCVWalking
Other              SCVOther
BurrowInit         [NONE]
ConstrctHarvst     [NONE]
IsWorking          SCVIsWorking
.headerend
# ----- #

SCVInit:
    imgul09        248 0 7 # SCVShad (terran\SCV.grp)
    playfram       0x00      # frame set 0
    goto           SCVOther

SCVOther:
    wait           125
    goto           SCVOther

SCVDeath:
    playsnd        369          # Terran\SCV\TSCDth00.WAV
    imgol08        332 0 0 # TerranBuildingExplosionsmall (thingy\tBangS.grp)
    wait           3
    end
```

```

SCVGndAttkInit:
    shvertpos    0
    wait         1
    playfram     0x22          # frame set 2
    attack25     1
    wait         1
    playfram     0x11          # frame set 1
    wait         1
    gotorepeatattk
    goto         SCVOther

SCVGndAttkToIdle:
    playfram     0x00          # frame set 0
    goto         SCVOther

SCVWalking:
    playfram     0x00          # frame set 0
    imgul08      249 0 0 # SCVGlow (thingy\tscGlow.grp)
    shvertpos    0
    goto         SCVOther

SCVIsWorking:
    shvertpos    0
    wait         1
local00:
    playfram     0x22          # frame set 2
    useweapon    14            # Fusion Cutter
    wait         1
    playfram     0x11          # frame set 1
    waitrand     8 10
    goto         local00

```

Its a quite a bit of stuff, but not overwhelming. Studying the header, I see that the initial animation of the SCV starts at the label 'SCVInit':

```

SCVInit:
    imgul09      248 0 7 # SCVShad (terran\SCV.grp)
    playfram     0x00          # frame set 0
    goto         SCVOther

SCVOther:
    wait         125
    goto         SCVOther

```

So, its animation starts by playing an image underlay (imgul09) which just means it plays another graphic under itself, the image being an images.dat number. The comment tells us that this is most probably its shadow. No problem there, we still want our SCV to see it's shadow. Then it plays a frame from frame set 0 (the SCV is a turning graphic, since it can face many directions), and then it goes to 'SCVOther', which is just below it. Here, it basically does nothing. It waits for a lot of ticks, loops, then waits for even more ticks. Looking at the other animations, we see that most end up here at 'SCVOther'. This looks like the SCV's idle animation. It waits, loops, and waits some more. Boring.

Let's make a little change:

```

SCVOther:
    shvertpos    1
    wait 1
    shvertpos    2

```

```

wait 1
shvertpos      3
wait 1
shvertpos      2
wait 1
goto           SCVOther

```

We add the 'shvertpos' (shift vertical position) instruction to make it turn move up 1 then 2 then 3 pixels and then move back down again, before looping. (You can think of the 'base' ground position at 0, and shvertpos moves the graphic up X pixels from the ground 0 position) Now our SCV should jump up and down while its idle. Cool.

After saving our text file, we need to compile it back into the original iscript.bin file. Now I select the **Compiler** tab. Then I click Add and select the scv.txt file that I just modified. I leave **Merge with default iscript.bin** checked because I just want to use my new SCV while leaving all the other units unchanged. Next I type in myiscript.bin in the **Save to:** field which indicates where my new iscript will be saved to. Finally, I click **Compile**. IceCC responds:

```

scv.txt:61: error: instruction 'd' not connected to a label; discarding
instructions until next label
scv.txt:61: error: unknown instruction opcode 'd'
icecc: error: aborted due to syntax errors in scv.txt

```

Oops. Looks like I accidentally typed a 'd' character on line 61. Didn't mean to do that. I go back and delete that and try **Compile** again. (Hint: if you select one of the error lines and then click on **Open in Editor**, IceCC will automatically open the offending file in the editor specified in the Preferences).

This time it works (no messages is a good sign; learn to love quietness :), and the output file is generated as myiscript.bin.

Now, I either fire up Stardraft, create a new CWAD, or use MPQ2K to create an MPQ and put myiscript.bin in:

```
scripts\iscript.bin
```

This is where you will always put your new iscript.bin file

Start up the patch loader (Stardraft for CWADs and MPQDraft for MPQs) and run my patch, and look at an SCV. Looks like we gave our peons a little too much sugar. :)

Of course, you can incorporate other files too. The iscript is just one part of the picture; advanced users will see how they can use it to make their own graphics move how they want them too, instead of following Starcraft's boring routines. More examples are in the `Examples` directory.

Appendix A: Animation Types

There are 28 different animations that entries can have. However, not all animation types use all of them (in fact, most only use the first 14 or less). They are:

```
0:  Init
1:  Death
2:  GndAttkInit
3:  AirAttkInit
4:  SpAbility1
5:  GndAttkRpt
6:  AirAttkRpt
7:  SpAbility2
8:  GndAttkToIdle
9:  AirAttkToIdle
10: SpAbility3
11: Walking
12: Other
13: BurrowInit
14: ConstrctHarvst
15: IsWorking
16: Landing
17: LiftOff
18: Unknown18
19: Unknown19
20: Unknown20
21: Unknown21
22: Unknown22
23: Unknown23
24: Unknown24
25: Burrow
26: UnBurrow
27: Unknown27
```

Init - this is the animation that plays when the graphic is 'spawned' (when it first appears on the screen). Generally, for unit graphics, this will play the shadow graphic as an underlay (so that the unit will have a shadow) and then go into its idle animation loop (which doesn't have a label name of its own, it just is kind of "in the middle" of everything).

Death - this is the animation that plays when the graphic is destroyed (e.g., when the unit dies). This will always end with the 'end' instruction, which terminates the script and tells Starcraft to remove the graphic.

GndAttkInit - this is the animation that plays when the unit the graphic is associated with begins to attack a target which is a ground unit. It generally end with a 'gotorepeatatk' instruction followed by a goto which goes back to its idle animation loop. This is because the unit may stop attacking, or it may continue attacking. If it is still attacking when it reaches the 'gotorepeatatk' instruction, it will goto the GndAttkRpt animation, otherwise it will go back to its idle animation. (If the 'gotorepeatatk' instruction is ommitted, the unit will only attack once, then stop) In the middle somewhere, it will contain a 'attack25' or 'attack26' or some other attack instruction which will tell it to actually do the damage or shoot its missile.

AirAttkInit - same, but for air units. Some graphics use the same animations for both air and ground attacks. This is fine, as most units look the same whether their target is air or ground.

SpAbility1 - this is main spell casting animation. It generally contains a 'castspell' instruction in the middle which is the actual point where the spell "takes effect." Most units don't use this and use the SpAbility2 animation instead.

GndAttkRpt - this is the animation that the graphic goes to if, after starting its attack, it continues to attack. Usually this is the same as the GndAttkInit animation (so its just a loop).

AirAttkRpt - same for the unit's air attack.

SpAbility2 - this is another spell casting animation. Some spells use this instead of the first (e.g., some units which have two different spells which use different animations).

GndAttkToIdle - this is the animation that plays when the unit stops attacking its ground target and returns to "idle." Generally this is a pointer to the unit's idle animation.

AirAttkToIdle - same for when the unit stops attacking an air target.

SpAbility3 - yet another spell casting animation. You'll have to look through the original scripts to see what units actually use this. Most don't.

Walking - this is the animation that plays when the unit moves from one location to another. Generally it will have some 'move' instructions which tells the graphic to move forward X pixels on the screen. Note that if the graphic is a unit graphic, and the flingy.dat entry that is associated with the animation set has its 'Move Control' variable set to something other than 'iscript control' (see a DAT editor like Arsenal III), then any 'move' instructions here will be ignored. For many flying units like the scourge, movement speed and acceleration is controlled by flingy.dat and not the actual animation. That is why when you look at the walking animation for the scourge it doesn't contain any 'move' instructions. Though it does contain the instructions to make the graphic bob up-and-down while it flies.

Other - Sort of Unknown. This is usually the "special" animation that is associated with very specific unit actions (it varies depending on what unit this sprite is associated with). For example, this is used for "suicide" attacks (like for the infested terran and scourge sprites).

BurrowInit - Some graphics can start out burrowed when they are spawned (e.g., Zerg units when a game begins, or are created with the 'Create Units with Properties' trigger). Instead of the Init animation playing, this one is played instead.

ConstructHarvst - This is the animation that plays when a building is being constructed (for the building) or when peons (SCVs, Drones, Probes) harvest minerals.

IsWorking - This is the animation for buildings which plays when the building is "working." E.G., when it is training a unit or upgrading an ability.

Landing - This is the animation that plays when a Terran building lands.

LiftOff - This is the animation that plays when a Terran building lifts off.

As you can see number 18-24 are still Unknown. They are rarely used, so you probably don't have to worry about them. If you want to help, you can try to figure out what they are for.

Burrow - This is the animation that plays when the unit burrows.

Unburrow - This plays when the unit unburrows.

Number 27 is also unknown.

There are several different 'Type' numbers which you can give to a header. Generally, you will keep the one that is already there. Each type contains a different number of animations, here are the ones that are known (see the beginning of this section for the tags the numbers refer to):

- Type 0 - Uses animations 0 to 1.
- Type 1 - Uses animations 0 to 1.
- Type 2 - Uses animations 0 to 3.
- Type 12 - Uses animations 0 to 13.
- Type 13 - Uses animations 0 to 13.
- Type 14 - Uses animations 0 to 15.
- Type 15 - Uses animations 0 to 15.
- Type 20 - Uses animations 0 to 21.
- Type 21 - Uses animations 0 to 21.
- Type 23 - Uses animations 0 to 23.
- Type 24 - Uses animations 0 to 25.
- Type 26 - Uses animations 0 to 27.
- Type 27 - Uses animations 0 to 27.
- Type 28 - Uses animations 0 to 27.
- Type 29 - Uses animations 0 to 27.

Remember, just because a set type contains a set of animations, doesn't mean it uses all of them. Any one can have a [NONE] label to signify that no animation is used (but make sure it really doesn't need it!). Take a browse through the iscript if you want to get an idea of what kinds of animation sets use which types. The pattern should be pretty intuitive.

Appendix B: Instruction Listfile.dat

This is a list of instructions which can be used in scripts. There are some that are unknown. This does not mean they will crash your script or anything (though they might), it just means that I haven't taken the time to test them. (These are usually recognized because their name is just usually just '_' prefixed to some hex number) If you would like to help, you can try these out and see what they do. To get a better idea, you can decompile the original iscript and see how it uses them. Each instruction may take a number of arguments. This is usually a couple of numbers (frame number to play, images.dat entry to use as an overlay graphic, etc.). For "jump" instructions like goto, this may also be a label (the label to "jump to"). A few examples:

Description: playfram <num>

Example: playfram 10

Description: goto <label>

Example: goto MyAnimation

Description: end

Example: end

Description: targtarcondjump <num> <num> <label>

Example: targtarcondjump 3 50 MyAnimation

There are also a few instructions which take a variable number of arguments. In this case, the first argument usually determines how many more there are. For example:

Description: playsndrand19 <var> <num> ...

Example: playsndrand19 1 10

Example: playsndrand19 3 12 13 15

Example: playsndrand19 10 1 2 3 4 5 6 7 8 9 10

In the description, it will have references to <1>, <2>, <3>, etc., where <#> refers to the argument number. Also, some instructions have a '0' character in them; this is a zero, not a capital Oh. There are a lot of them (over 60, but there are only a few that you'll use frequently, like **playfram**, **wait**, **goto**, some of the **imgol*** and **sprol*** ones to spawn overlay/underlay graphics, the **playsnd*** instructions to play sounds, and occasionally the **attack***, **castspell**, and **move** instructions).

playfram <num> - Play frame <1>: Pretty basic. Play frame number <1> for the main sprite (i.e., frame number in the GRP). If the images.dat entry this animation header is associated with has "turn gfx" checked, then this instruction will play frame <1> plus a number up to 17 and possibly mirroring the frame image, depending on which direction the sprite is facing. (That's why GRP frames are in sets of 17) Otherwise it will play frame <1> specifically no matter what direction the sprite is facing. ICE: If you double click on one of these instructions it will show the actual frame of the original GRP in the Preview GRP tab.

playframfile <num> - Play frame <1> according to tileset: Slightly unknown. It is used for sprites like the vespe geysers, plays a different frame depending on what tileset this is. Not entirely sure how it is calculated.

__02 <num> - UNKNOWN OP: 0x02.

shvertpos <num> - Shift graphic position <1> vertical pixels: Shift the graphic <1> pixels downward from its current position. (Doesn't change frame, just move the current image down) It will remain shifted down <1> pixels until you shift it again; e.g., to shift it back to its original "home" position, use this instruction to shift it with %1 = 0.

__04 <num> - UNKNOWN OP.

wait <num> - Wait <1> ticks: Wait for <1> tenths of a second before moving on.

waitrand <num> <num> - Wait either <1> or <2> ticks (random): Randomly wait one of 2 times: either wait <1> tenths of a second or wait <2> tenths of a second.

goto <label> - Go to offset <1>: Go to offset <1> and continue animating from there. ICE: Double Click on one of these instructions to jump to offset %1. (You can press back to get back to the original script)

imgol08 <num> <num> <num> - Place active overlay (images.dat) <1> at vertical offset <2>, horizontal offset <3>: This instruction spawns an "active" overlay graphic from images.dat entry number <1> on top of the current sprite at a vertical offset of <2> pixels, and horizontal <3>. An "active" overlay is one that follows the main graphic around (like a shadow graphic). By "spawning" an images.dat graphic, I mean exactly that: a images.dat graphic just like any other images.dat graphic is created. It has its own palette settings, gfx settings, and even its own iscript.bin animation, just like any other images.dat entry. ICE: You can double click on one of these instructions to display the GRP associated with images.dat entry <1> in the Secondary GRP Preview window.

imgul09 <num> <num> <num> - Place active underlay (images.dat) <1> at vertical offset <2>, horizontal <3>: Same as imgol08 but underlay instead of an overlay. (Overlay means over the current graphic, underlay means underneath the current graphic)

imgol0a <num> - Unknown. <1> is a reference to images.dat like imgol08 and imgul09.

switchul <num> - Switch current underlay for underlay (images.dat) <1>. Predict to do with powerups: Basically unknown.

__0c - UNKNOWN OP.

imgol0d <num> <num> <num> - Place overlay (images.dat) <1> at vertical offset <2>, horizontal <3>. Requires LOG file: This will spawn a certain independent overlay images.dat entry <1> at a vertical offset of <2> units. However, this instruction also requires a LO* file associated with the images.dat entry the main graphic is associated with (the LO* file gives it additional coordinates for placing the overlay). Sort of unknown so only mess with it if you know what you're doing.

imgol0e <num> <num> <num> - Like the images overlays above. I haven't tested this one.

sprol0f <num> <num> <num> - Place independent overlay (sprites.dat) <1> at vertical offset <2>, horizontal <3>: This will spawn an independent overlay sprites.dat entry number <1> at a vertical offset of <2> pixels and horizontal <3>. An independent overlay is different from an active overlay in that it does not follow the main sprite around (it is created and then treated as its own sprite, "independent" of what the main sprite does). Usually used for showing explosions on top of a unit that just died.

sprol10 <num> <num> <num> - Place independent overlay (sprites.dat) <1>, over everything below this anim level, at vertical offset <2>, horizontal <3>: This is just like sprol0f, but instead of spawning the sprite at an animation level just above the current main sprite, it is spawned on top of all sprites (top animation level).

sprul11 <num> <num> <num> - Place independent underlay (sprites.dat) <1> at vertical offset <2>, horizontal <3>: Just like above, but an underlay instead of an overlay graphic. (Underneath all sprites, the lowest animation level) Usually used for displaying "rubble" or dead bodies sprites after a unit is killed.

__12 - UNKNOWN OP.

sprol13 <num> <num> <num> - Display overlay (sprites.dat) <1> with vertical offset <2>, horizontal <3>.

Requires LO* file: Sort of Unknown. Requires additional coordinates from a LO* file.

sproll14 <num> <num> <num> - Like the above. Haven't tried this one.

sproll15 <num> <num> - Display overlay (sprites.dat) <1> with unknown var <2>. Requires LOG file: Sort of Unknown. Basically a spawn sprites.dat graphic as an independent overlay, but requiring coordinates from a LO* file.

end - End animation, remove graphic: Just what it says, end the animation and remove the sprite. It will never come back ever again. :|

__17 <num> - UNKNOWN OP.

playsnd <num> - Play sound (sfxdata.dat) <1>: Play a sound entry from sfxdata.dat, entry number <1>.

playsndrand <var> <num> ... - Randomly play one of <1> sounds (sfxdata.dat): <2> or <3> or <4> or...: This is like playsnd but will play one of several sounds at random. For example, say you have 2 sounds you might want to play (though only one of them will actually be played), then <1> would be 2. And then the next variables <2> and <3> would be the actual sfxdata.dat entry numbers of your sounds. If you had 3 sounds to play at random, you would have one more variable <4>. Etc.

playsndbtwn <num> <num> - Randomly play a sound (sfxdata.dat) between entries <1> and <2>: Just like playsnd and playsndrand, but play a sound from sfxdata.dat randomly between entry <1> and <2> (inclusive). So if you had <1> as 10 and <2> as 20, then this instruction randomly plays sound 10, 11, 12, ..., 19, or 20.

domissiledmg - Do damage (missiles): This instruction is specifically for missile sprites in their attack animations. Basically, at this point it will do damage to its target.

attack1c <var> <num> ... - Attack with the appropriate weapon (ground weapon if in the ground attack animation, air for the air attack) and randomly choose between <1> sounds <2>, <3>, <4>, etc. to play: This instruction is used in attack animations. Basically, it makes the unit associated with this sprite "attack" (do damage or shoot out a missile). Its basically just playsndrand along with attack25 combined in 1 instruction.

followmaingraphic - Follow frame changes of main graphic (underlay instruction for shadows): This instruction is used in the initial animations of certain sprites which are normally called as active overlay or underlay graphics by other sprites (like shadows). It informs the sprite to "follow the frame changes" of the main graphic it is an overlay/underlay for. E.G., say this is on a shadow sprite and the shadow is spawned as an active underlay by its main unit sprite. When the unit sprite plays frame 1, this overlay/underlay will also play frame 1, when it plays frame X, this will also play frame X. Etc.

__1e_condjump <num> <label> - Using randomizer value <1>, jump to label <2>: Basically, generate a random number (somehow related to the randomizer variable <1>) and then if that random number is within a certain range, then goto label <2>. Otherwise just skip this instruction. This is known as a "conditional jump" since it is like a 'goto', but you only take the jump if a certain condition is fulfilled (in this case, the randomizer variable is correct at the time). This is useful to get randomized animation routines playing (such as the marine and firebat's gun pointing and turning animations when idle).

turnccwise <num> - Turn graphic <1> frames counterclockwise: Change the direction the current sprite is facing by turning it <1> frames counterclockwise (in more technical terms, that means play the frame that is <1> backward from the current frame in the GRP, and maybe mirrored).

turnwise <num> - Turn graphic <1> frames clockwise: Same as turnccwise but turn frames clockwise (forward in the GRP frameset).

turn1cwise - Turn graphic 1 frame clockwise: Same as above but specifically only turn 1 frame.

turnrand <num> - Turn graphic <1> frames in a random direction: Same as above, but turn randomly either clockwise or counterclockwise.

__23 <num> - UNKNOWN OP.

sigorder <num> - (?) Send signal to unit for order: <1>: This is a very important instruction, but I'm really not very sure what it does. Essentially, it sends some sort of signal (variable <1>) to the game which tells it something. But I'm not sure what that is. If you see this in one of the original animations you are editing, it may be a good idea to keep it there in some place or another. (my guess is that it has to do with spell casting)

attack25 <num> - Attack with <1> (1 = Ground, else Air): This instruction is used in attack animations. Basically, it makes the unit associated with this sprite "attack" (do damage or shoot out a missile). If <1> is 1 then it will attack with its ground weapon (specified in units.dat) otherwise it will attack with its air weapon. You can "attack" more than once in an animation to do multiple damage/shots (a'la the valkyrie).

attack26 - Attack with appropriate weapon: Same as attack25, but you don't have to specify the weapon. (It will attack with the ground weapon if attacking a ground unit, otherwise attack with the air weapon)

castspell - Cast Spell: Used in spell casting animations (specifically the Special Ability Animations usually). At the point when this instruction is called, the actual spell or special ability will be "cast" (shoot the weapon or start whatever action associated with the spell).

useweapon <num> - Use weapon (weapons.dat) <1>: This instruction forces the unit associated with this sprite to "attack" its target with weapon number <1> in weapons.dat. Normally this is used in the harvesting animation of peons to make them "attack" minerals with their harvesting weapon (same as their normal weapon actually). But it works just fine in other animations which assume the unit has a target. (e.g., you can make the SCV use the Psi Storm weapon every time it tries to harvest something for some fun :)

move <num> - Move graphic <1> units forward: Used specifically in the walking animation to move the sprite <1> units forward in the direction it is facing.

gotorepeatattk - Goto Repeated Attack if still attacking: It is believed that this instruction tells the sprite where it should continue looping an attack animation if its target is still alive. For example, you would place this at the end of the Initial and Repeated Attack animations. When the sprite gets to this point in the animation, it checks to see if its target is still alive and then goes to the Repeated Attack Animation if it still is (otherwise it goes to the Return to Idle from Ground/Air Attack). If this instruction is not present, the unit will not loop its attack.

__2b <num> - UNKNOWN OP.

__2c - UNKNOWN OP.

__2d - UNKNOWN OP.

nobrkcodestart - Begin unbreakable code section: This instruction is paired with nobrkcodeend (you start an unbreakable section of code with nobrkcodestart and end it with nobrkcodeend). When a sprite entered an "unbreakable" section of code, it will not be able to carry out any other order given to it by the player except for special canceling orders like stop (e.g., during a battle cruiser's yamato gun shot). This is used in many attack animations to "enclose" the start an end of the attacking animation (because otherwise players could manipulate the animation and only let the part of the anim play through until the "attack" and then skip the rest and start over

again).

nobrkcodedeend - End unbreakable code section: See previous.

ignorereest - (?) Ignore other instructions: Unknown. Appears to cause the rest of the instructions in this routine be ignored (except for goto instructions).

attkprojangle <num> - Attack with projectile spawned from angle <1>: Sort of Unknown. This causes the unit to attack with its appropriate weapon (if it is a projectile or missile weapon) like attack25 but the missile will appear from an angle designated by <1>.

tmprmgraphicstart - Temporarily remove graphic: This will cause the sprite to be "temporarily removed" until it comes back with the tmprmgraphicend instruction (see next). "Temporarily removed" means it becomes invisible.

tmprmgraphicend - Play graphic again after temporarily removed (by previous): This instruction again makes a sprite visible after being removed by tmprmgraphicstart.

playframno <num> - Play specific frame <1>: This instruction is like playfram but it will always play a specific frame number <1> (even if the images.dat entry associated with this animation header has gfx turns checked). I.E., it will not add upto 17 frames or mirror the graphic depending on what direction it is facing.

__35_condjump <label> - If (?) unknown then goto <1>: Unknown.

__36 - UNKNOWN OP.

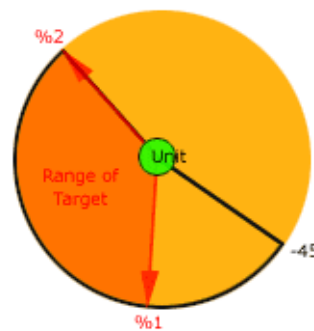
__37 - UNKNOWN OP.

__38 <num> - UNKNOWN OP. Predict something to do with LOS file

pwrupcondjump <label> - If picked up (powerup) goto label <1>: Sort of unknown. Normally is instruction is in the initial animations of powerups. When a powerup is picked up, it is kind of "spawned again" in the hands of the peon. This instruction tells the animation of the powerup to go to label <1> if it is picked up (as opposed to on the ground). If it isn't picked up, this is ignored.

trgtrangecondjump <num> <label> - If target within range of <1> (pixels?) goto label <2>: This is used in certain attack animations (such as the regular archon's). Basically, when the animation gets to this point, if the sprite's target is within range of <1> pixels (maybe different unit of measurement), then goto offset <2>. Otherwise if the target is not within range then ignore this instruction. This is useful if you want a unit (like the archon) to have different attack animations for units that are at different distances away (at different distances, for example, the archon's energy attack has a different length tail). You can use a little logic and combine these as well for attacks at many different distances (e.g., If target within 5 then goto X, else if unit is within 10 goto Y, else if unit is within 15 goto Z, else... etc.).

trgtarccondjmp <num> <num> <label> - If target in arc with starting angle at <1> clockwise from -45 degrees, until <2> degrees, goto label <3>: This instruction is a bit complicated. It is normally used for attack animations for units such as the sunken colony. It helps if I draw a picture (which may be a bit off, but you'll get the idea).



This instruction goes to a certain label <3> if the current sprite's target is within the range of angle <1> and <2>. Otherwise it will ignore this instruction. The picture illustrates the best. What this is useful for is if you have a unit which maybe stationary (like the sunken colony), but you want it to have different attack animations depending on which direction its target is in.

__3c_condjmp <label> - UNKNOWN OP. Some kind of conditional jump.

__3d <num> - UNKNOWN OP.

__3e - UNKNOWN OP.

__3f_condjmp - If (?) unknown go to label <1>: Unknown.

__40 <num> - UNKNOWN OP.

__41 <num> - UNKNOWN OP.

sprol42 <num> - <1> refers to a sprites.dat entry. Other than that, I'm not sure what this does.

__43 - UNKNOWN OP.

__44 - UNKNOWN OP.

Remember, the best way to learn about instructions is to actually decompile the iscript and look at the original animations yourself to see how the original Starcraft programmers used them. If you want to know how a certain graphic does this or that, just look at its header set and it should become clear.

Appendix C: Script File Specification

This section provides a quick reference to make sure you have your script syntax right. A script file may contain any number of headers. Each must have a unique ID (the `IsId` variable). The headers may be placed anywhere: all at the beginning, all at the end, or interspersed between animation scripts. The decompiler puts them just above the animation scripts they use. In addition to headers, the file contains scripts, which consist of labels and series of instructions. In general, the general outline is free format (spaces and tabs can be used freely), and usually the compiler will figure out what you're up to. All syntax is case sensitive.

The header format is the following:

```
.headerstart
<header variable> <value>
<header variable> <value>
<header variable> <value>
...
.headerend
```

The header variables are the two special variables `IsId` and `Type` and the animation tags (see Appendix A). They can be in any order. `IsId` is followed by a number (the iscript ID) and `Type` is followed by a number (the set type, see Appendix A). Each animation tag is followed by the label it starts at or the special tag `[NONE]` for no animation.

Legal label may use any characters except for colons (`:`), hashes (`#`), spaces, tabs, or newlines. They also may not be the special tags `.headerstart`, `.headerend`, or `[NONE]`. A label must always be placed before a valid instruction. It is declared with its name followed by a colon (`:`); both the name and colon must be on the same line. It does not matter whether the instruction is on the same line, or some lines below, but a header is not allowed to be in between. E.G., all the following are legal:

```
label: instruction
```

```
label:
instruction
```

```
label :

    instruction
```

Which style you use is up to you. Again, spaces and tabs are free form: you can put a tab before the instruction and tab after, or just use spaces, or a combination; the compiler doesn't care. Whatever works for you. The label is said to "point to" the instruction which follows it. Any reference to that label will effectively be a "jump" to that instruction. You may declare multiple labels which point to the same instruction. This is done by placing multiple label declarations immediately after one another. E.G.,

```
label1: label2: label3:
label4:
    instruction
```

All four labels point to the same instruction. Each label may only be declared once in a file (it may be *used* by as many headers and/or jump instructions as you like, but you may not have two labels with the same name which point to different instructions).

Each instruction must be separated by at least one line (i.e., no two instructions may be on the same line). All the arguments to the instruction should follow it on the same line. While they should be separated by white space (tabs or spaces), again, it is free form, so format however you like. Examples of legal uses:

```
instr 0 10
instr2      0
instr4      10    1 2 3 4 5 6 7 8 9 10
            instr5 label
            instr3

            instr6 10 20          label
```

Instruction arguments are either numbers or label names (see Appendix B for individual formats). Number values are normally interpreted as decimal numbers (base 10) unless they are prefixed by '0x' in which case they are interpreted as a hexadecimal number (base 16) or if they are prefixed by one or more '0's (zeros) in which case they are interpreted as octal (base 8).

Two instructions are "connected" (that is, the later will run after the former) if the former one is not a 'goto' or an 'end' instruction. You may place a header declaration in between two connected instructions, but it is not suggested (the compiler will warn you). The last instruction in the file must always be terminal (either a goto or end).

Comments begin with a hash (#) character. They may begin anywhere on a line and everything after the hash is ignored by the compiler. All the following are legal:

```
# this is a comment
.headerstart      # this starts a header
IsId  10          # this the iscript ID
Type  0           # this is the set type
Init  [NONE]# none
Death foo        # death uses label foo
.headerend

foo:              # this is a label
    wait 125      # wait a long time!
    goto foo      # loop
```

No voodoo about it.

Appendix D: Other Useful Programs

No Starcraft program is useful on its own. For IceCC, you will probably find yourself trying to look up what all those numbers are all the time (whether it be images.dat entries, sprites.dat entries, or whatever). As a starter, entry lists of the dat files have been provided in the "help" directory of IceCC. But you will probably also want to check out these programs if you haven't already:

Stardraft (<http://www.camsys.org>) - to patch Starcraft of course.

MPQ2k (<http://www.campaigncreations.com/starcraft/mpq2k>) - to extract, import files into the MPQ.

Arsenal III/ArrEdit (<http://www.camsys.org>) - A DAT editor, viewer. You can use Arsenal to gather information about what number corresponds to what DAT entry.

Arsenal ZERO (<http://magnus99.dhs.org/downloads/>) - a toned down command line DAT editor. It will extract entries from a DAT to a text file for easy browsing.

Retro GRP (<http://www.infoceptor.com/files/starcraft.shtml>) - a GRP (the collection of bitmap image frames) viewer and editor. You will probably want to look at the actual frames of the graphic when using those 'playfram' instructions.

ICE (<http://www.camsys.org>) - If IceCC is too hard-core for you. :) But ICE is still useful in conjunction with IceCC, since you can use it to preview images and sound effects.

MPQView (<http://www.starcraft.org/downloads/downloadutils.shtml>) - MPQ2k can be used to extract files from an MPQ also, but MPQView is a graphical tool that makes it painless and allows you to browse the file listing more easily.

Text Editors (http://www.thefreesite.com/Free_Software/Free_text_editors/index.html) - Notepad too basic for ya? Here's some free text editors that are more advanced.

Mac OS X Starcraft programs can be found at <http://www.macstorm.net> and <http://customs.macstarcraft.com/>.

Appendix E: Program Configuration

I have yet to explain the '-c' and '-r' options which you can give to both icecc and icedc.

-c configfile

This option allows you to specify a configuration file to use besides the default (which is C:\IceCC\icecc.ini). The only thing the config file is the line:

```
INSTALLDIR=C:\IceCC
```

If you installed IceCC somewhere else, you should change the path after the '='. You can also use the variable 'CONFIGDIR=*somedirpath*' to change the default configuration files directory (see below). Right now, this option is not all that useful.

-r configdir

You'll notice that IceCC uses lots of other files besides the iscript.bin file. For example, it uses images.dat and images.tbl files to determine which iscript IDs are associated to which images.dat entries and to print the GRP file path to the graphics file in a comment. By default, the files it uses are in the "data" directory. If you are making a large Total Conversion and want it to use your own custom DAT, TBL, and other files instead, you can make your own data directory (the path format of each file is the same as in the MPQ) with your custom files and have icecc read that instead. To see which files you are required to have, look in the original data dir; you can just make a copy of it and just replace the files you have replacements for. (The *.lst files are not actual starcraft files, but are DAT entry listings with names; Each name is on a separate name and they should not contain spaces) This is really for advanced users and you probably will never even bother with this.

If you are a programmer, you may want to look at the source code to do some more customization. It has lots of routines for dealing with Starcraft data files, just in case you want to use them in your own project. IceCC is written in ANSI C, and nothing else (i.e., it was done "from scratch" without any tool kits or anything). It was also written on Linux. For all the things Tux still lacks, it is far and away the best development platform; I can even play Starcraft to test my code on it. :) IceCC is licensed under the GNU General Public License and thus is "open source." You can download the source at <http://magnus99.dhs.org/downloads>. The source code to this build is named 'icecc-*-src.zip' or 'icecc-*-src.tar.gz' where * is the version number. The latest build (warning: may not be stable!) will be named 'icecc-*.zip' where * is the date.

Appendix F: DAT Entry Lists

The iscript makes many references to DAT file entries (e.g., the sounds to play come from sfxdata.dat, the images to display come from images.dat and sprites.dat, etc.). Unfortunately IceCC shows these references as numbers so they don't have much meaning unless you can remember the several thousand things in your head. These lists help you track down what the numbers refer to. You can also use Arsenal 3 or Arsenal ZERO.

Units.DAT:

0=Terran Marine	65=Protoss Zealot	130=Infested Command Center
1=Terran Ghost	66=Protoss Dragoon	131=Zerg Hatchery
2=Terran Vulture	67=Protoss High Templar	132=Zerg Lair
3=Terran Goliath	68=Protoss Archon	133=Zerg Hive
4=Goliath Turret	69=Protoss Shuttle	134=Zerg Nydus Canal
5=Terran Siege Tank (Tank Mode)	70=Protoss Scout	135=Zerg Hydralisk Den
6=Tank Turret (Tank Mode)	71=Protoss Arbiter	136=Zerg Defiler Mound
7=Terran SCV	72=Protoss Carrier	137=Zerg Greater Spire
8=Terran Wraith	73=Protoss Interceptor	138=Zerg Queen's Nest
9=Terran Science Vessel	74=Dark Templar	139=Zerg Evolution Chamber
10=Gui Montang (Firebat)	75=Zeratul (Dark Templar)	140=Zerg Ultralisk Cavern
11=Terran Dropship	76=Tassadar/Zeratul (Archon)	141=Zerg Spire
12=Terran Battlecruiser	77=Fenix (Zealot)	142=Zerg Spawning Pool
13=Vulture Spider Mine	78=Fenix (Dragoon)	143=Zerg Creep Colony
14=Nuclear Missile	79=Tassadar (Templar)	144=Zerg Spore Colony
15=Terran Civilian	80=Mojo (Scout)	145=Unused Zerg Building
16=Sarah Kerrigan (Ghost)	81=Warbringer (Reaver)	146=Zerg Sunken Colony
17=Alan Schezar (Goliath)	82=Gantrithor (Carrier)	147=Zerg Overmind (With Shell)
18=Alan Schezar Turret	83=Protoss Reaver	148=Zerg Overmind
19=Jim Raynor (Vulture)	84=Protoss Observer	149=Zerg Extractor
20=Jim Raynor (Marine)	85=Protoss Scarab	150=Mature Chrysalis
21=Tom Kazansky (Wraith)	86=Danimoth (Arbiter)	151=Zerg Cerebrate
22=Magellan (Science Vessel)	87=Aldaris (Templar)	152=Zerg Cerebrate Daggoth
23=Edmund Duke (Tank Mode)	88=Artanis (Scout)	153=Unused Zerg Building 5
24=Edmund Duke Turret (Tank Mode)	89=Rhynadon (Badlands Critter)	154=Protoss Nexus
25=Edmund Duke (Siege Mode)	90=Bengalaas (Jungle Critter)	155=Protoss Robotics Facility
26=Edmund Duke Turret (Siege Mode)	91=Unused - Was Cargo Ship	156=Protoss Pylon
27=Arcturus Mengsk (Battlecruiser)	92=Unused - Was Mercenary Gunship	157=Protoss Assimilator
28=Hyperion (Battlecruiser)	93=Scantid (Desert Critter)	158=Unused Protoss Building (158)
29=Norad II (Battlecruiser)	94=Kakaru (Twilight Critter)	159=Protoss Observatory
30=Terran Siege Tank (Siege Mode)	95=Ragnasaur (Ashworld Critter)	160=Protoss Gateway
31=Tank Turret (Siege Mode)	96=Ursadon (Ice World Critter)	161=Unused Protoss Building (162)
32=Firebat	97=Lurker Egg	162=Protoss Photon Cannon
33=Scanner Sweep	98=Raszagal	163=Protoss Citadel of Adun
34=Terran Medic	99=Samir Duran (Ghost)	164=Protoss Cybernetics Core
35=Zerg Larva	100=Alexei Stukov (Ghost)	165=Protoss Templar Archives
36=Zerg Egg	101=Map Revealer	166=Protoss Forge
37=Zerg Zergling	102=Gerard DuGalle	167=Protoss Stargate
38=Zerg Hydralisk	103=Zerg Lurker	168=Stasis Cell/Prison
39=Zerg Ultralisk	104=Infested Duran	169=Protoss Fleet Beacon
40=Zerg Broodling	105=Disruption Web	170=Protoss Arbiter Tribunal
41=Zerg Drone	106=Terran Command Center	171=Protoss Robotics Support Bay
42=Zerg Overlord	107=Terran Comsat Station	172=Protoss Shield Battery
43=Zerg Mutalisk	108=Terran Nuclear Silo	173=Khaydarin Crystal Formation
44=Zerg Guardian	109=Terran Supply Depot	174=Protoss Temple
45=Zerg Queen	110=Terran Refinery	175=Xel'Naga Temple
46=Zerg Defiler	111=Terran Barracks	176=Mineral Field (Type 1)
47=Zerg Scourge	112=Terran Academy	177=Mineral Field (Type 2)
48=Torrarsque (Ultralisk)	113=Terran Factory	178=Mineral Field (Type 3)
49=Matriarch (Queen)	114=Terran Starport	179=Cave
50=Infested Terran	115=Terran Control Tower	180=Cave-in
51=Infested Kerrigan	116=Terran Science Facility	181=Cantina
52=Unclean One (Defiler)	117=Terran Covert Ops	182=Mining Platform
53=Hunter Killer (Hydralisk)	118=Terran Physics Lab	183=Independant Command Center
54=Devouring One (Zergling)	119=Unused - Was Starbase?	184=Independant Starport
55=Kukulza (Mutalisk)	120=Terran Machine Shop	185=Independant Jump Gate
56=Kukulza (Guardian)	121=Unused - Was Repair Bay?	186=Ruins
57=Yggdrasil (Overlord)	122=Terran Engineering Bay	187=Kyadarin Crystal Formation
58=Terran Valkyrie Frigate	123=Terran Armory	188=Vespene Geyser
59=Mutalisk/Guardian Cocoon	124=Terran Missile Turret	189=Warp Gate
60=Protoss Corsair	125=Terran Bunker	190=PSI Disruptor
61=Protoss Dark Templar	126=Norad II	191=Zerg Marker
62=Zerg Devourer	127=Ion Cannon	192=Terran Marker
63=Protoss Dark Archon	128=Uraj Crystal	193=Protoss Marker
64=Protoss Probe	129=Khalis Crystal	194=Zerg Beacon

195=Terran Beacon
 196=Protoss Beacon
 197=Zerg Flag Beacon
 198=Terran Flag Beacon
 199=Protoss Flag Beacon
 200=Power Generator
 201=Overmind Cocoon
 202=Dark Swarm
 203=Floor Missile Trap
 204=Floor Hatch
 205=Left Upper Level Door

206=Right Upper Level Door
 207=Left Pit Door
 208=Right Pit Door
 209=Floor Gun Trap
 210=Left Wall Missile Trap
 211=Left Wall Flame Trap
 212=Right Wall Missile Trap
 213=Right Wall Flame Trap
 214=Start Location
 215=Flag
 216=Young Chrysalis

217=Psi Emitter
 218=Data Disc
 219=Khaydarin Crystal
 220=Mineral Cluster Type 1
 221=Mineral Cluster Type 2
 222=Protoss Vespene Gas Orb Type 1
 223=Protoss Vespene Gas Orb Type 2
 224=Zerg Vespene Gas Sac Type 1
 225=Zerg Vespene Gas Sac Type 2
 226=Terran Vespene Gas Tank Type 1
 227=Terran Vespene Gas Tank Type 2

Flingy.DAT:

0=Scourge
 1=Broodling
 2=Infested Terran
 3=Guardian Cocoon
 4=Defiler
 5=Drone
 6=Zerg Egg
 7=Guardian
 8=Hydralisk
 9=Infested Kerrigan
 10=Larva
 11=Mutalisk
 12=Overlord
 13=Queen
 14=Ultralisk
 15=Zergling
 16=Cerebrate
 17=Infested Command Center
 18=Spawning Pool
 19=Mature Chrysalis
 20=Evolution Chamber
 21=Creep Colony
 22=Hatchery
 23=Hive
 24=Lair
 25=Sunken Colony
 26=Greater Spire
 27=Defiler Mound
 28=Queen's Nest
 29=Nydus Canal
 30=Overmind w/shell
 31=Overmind w/out shell
 32=Ultralisk Cavern
 33=Extractor
 34=Hydralisk Den
 35=Spire
 36=Spore Colony
 37=Arbiter
 38=Archon Energy
 39=Carrier
 40=Dragoon
 41=Interceptor
 42=Probe
 43=Scout
 44=Shuttle
 45=High Templar
 46=Dark Templar (Hero)
 47=Reaver
 48=Scarab
 49=Zealot
 50=Observer
 51=Templar Archives
 52=Assimilator
 53=Observatory
 54=Citadel of Adun
 55=Forge
 56=Gateway
 57=Cybernetics Core
 58=Khaydarin Crystal
 59=Nexus
 60=Photon Cannon
 61=Arbiter Tribunal
 62=Pylon

63=Robotics Facility
 64=Shield Battery
 65=Stargate
 66=Stasis Cell/Prison
 67=Robotics Support Bay
 68=Protoss Temple
 69=Fleet Beacon
 70=Battlecruiser
 71=Civilian
 72=Dropship
 73=Firebat
 74=Ghost
 75=Goliath Base
 76=Goliath Turret
 77=Kerrigan (Ghost)
 78=Marine
 79=Unknown 19
 80=Wraith
 81=SCV
 82=Siege Tank (Tank) Base
 83=Siege Tank (Tank) Turret
 84=Siege Tank (Siege) Base
 85=Siege Tank (Siege) Turret
 86=Science Vessel
 87=Science Vessel Turret
 88=Vulture
 89=Spider Mine
 90=Academy
 91=Barracks
 92=Armory
 93=Comsat Station
 94=Command Center
 95=Supply Depot
 96=Control Tower
 97=Factory
 98=Covert Ops
 99=Ion Cannon
 100=Machine Shop
 101=Missile Turret (Base)
 102=Crashed Norad II
 103=Physics Lab
 104=Bunker
 105=Refinery
 106=Barracks
 107=Science Facility
 108=Nuke Silo
 109=Unknown 20
 110=Starport
 111=Unknown 22
 112=Unknown 23
 113=Unknown 24
 114=Ragnasaur (Ashworld)
 115=Rynadon (Badlands)
 116=Bengalass (Jungle)
 117=Vespene Geyser
 118=Mineral Deposit 1
 119=Mineral Deposit 2
 120=Mineral Deposit 3
 121=Unused
 122=Zerg Beacon
 123=Terran Beacon
 124=Protoss Beacon
 125=Dark Swarm

126=Flag
 127=Chrysalis
 128=Psi Emmitter
 129=Data Disc
 130=Khadarin Crystal
 131=Mineral Chunk
 132=Unknown 28
 133=ProtossGas Orb
 134=Unknown 29
 135=Zerg Gas Sac
 136=Unknown 30
 137=Terran Gas Tank
 138=Unknown 31
 139=Unknown 33
 140=Start Location
 141=Fusion Cutter
 142=Unknown 67
 143=Unknown 71
 144=Gemini Missles
 145=Unknown 68
 146=Unknown 70
 147=Unknown 69
 148=Unknown 72
 149=Unknown 73
 150=Unknown 74
 151=Unknown 77
 152=Unknown 62
 153=Unknown 63
 154=Unknown 64
 155=Unknown 65
 156=Unknown 89
 157=Psi Storm
 158=Unknown 75
 159=Phase Disruptor
 160=Unknown 66
 161=Unknown 80
 162=Unknown 81
 163=Unknown 82
 164=Unknown 79
 165=Unknown 83
 166=Unknown 84
 167=Unknown 85
 168=Unknown 87
 169=Consume
 170=Ensnare
 171=Unknown 78
 172=Unknown 92
 173=Unknown 101
 174=Unknown 102
 175=Unknown 103
 176=Unknown 104
 177=Unknown 105
 178=Unknown 34
 179=Unknown 39
 180=Unknown 35
 181=Unknown 36
 182=Unknown 37
 183=Unknown 38
 184=Unknown 201
 185=Unknown 202
 186=Unknown 205
 187=Unknown 206
 188=Unknown 208

189=Unknown 209
 190=Unknown 211
 191=Unknown 207
 192=Unknown 223
 193=Unknown 215
 194=Unknown 218
 195=Unknown 219

196=Unknown 216
 197=Unknown 217
 198=Unknown 212
 199=Unknown 213
 200=Unknown 214
 201=Unknown 228
 202=Unknown 229

203=Unknown 230
 204=Unknown 232
 205=Unknown 233
 206=Unknown 226
 207=Unknown 235
 208=Unknown 236

Sprites.DAT:

0=Ash World Doodad (Rock 1)
 1=Ash World Doodad (Rock 2)
 2=Ash World Doodad (Rock 3)
 3=Ash World Doodad (Rock 4)
 4=Ash World Doodad (Rock 5)
 5=Ash World Doodad
 6=Ash World Doodad
 7=Ash World Doodad
 8=Ash World Doodad
 9=Ash World Doodad
 10=Ash World Doodad
 11=Ash World Doodad
 12=Ash World Doodad
 13=Ash World Doodad
 14=Ash World Doodad
 15=Ash World Doodad
 16=Jungle Doodad
 17=Jungle Doodad
 18=Jungle Doodad
 19=Jungle Doodad
 20=Jungle Doodad
 21=Jungle Doodad
 22=Jungle Doodad
 23=Jungle Doodad
 24=Jungle Doodad
 25=Jungle Doodad
 26=Jungle Doodad
 27=Jungle Doodad
 28=Jungle Doodad
 29=Jungle Doodad
 30=Jungle Doodad
 31=Jungle Doodad
 32=Jungle Doodad
 33=Jungle Doodad
 34=Jungle Doodad
 35=Jungle Doodad
 36=Jungle Doodad
 37=Jungle Doodad
 38=Jungle Doodad
 39=Jungle Doodad
 40=Jungle Doodad
 41=Jungle Doodad
 42=Jungle Doodad
 43=Jungle Doodad
 44=Jungle Doodad
 45=Jungle Doodad
 46=Jungle Doodad
 47=Jungle Doodad
 48=Jungle Doodad
 49=Jungle Doodad
 50=Jungle Doodad
 51=Jungle Doodad
 52=Jungle Doodad
 53=Jungle Doodad
 54=Jungle Doodad
 55=Jungle Doodad
 56=Jungle Doodad
 57=Space Platform Doodad
 58=Space Platform Doodad
 59=Space Platform Doodad
 60=Space Platform Doodad
 61=Space Platform Doodad
 62=Space Platform Doodad
 63=Space Platform Doodad
 64=Space Platform Doodad
 65=Space Platform Doodad
 66=Space Platform Doodad

67=Space Platform Doodad
 68=Space Platform Doodad
 69=Space Platform Doodad
 70=Space Platform Doodad
 71=Space Platform Doodad
 72=Space Platform Doodad
 73=Space Platform Doodad
 74=Space Platform Doodad
 75=Space Platform Doodad
 76=Badlands Doodad
 77=Badlands Doodad
 78=Badlands Doodad
 79=Badlands Doodad
 80=Badlands Doodad
 81=Badlands Doodad
 82=Badlands Doodad
 83=Badlands Doodad
 84=Badlands Doodad
 85=Badlands Doodad
 86=Badlands Doodad
 87=Badlands Doodad
 88=Badlands Doodad
 89=Badlands Doodad
 90=Badlands Doodad
 91=Badlands Doodad
 92=Badlands Doodad
 93=Badlands Doodad
 94=Badlands Doodad
 95=Badlands Doodad
 96=Badlands Doodad
 97=Badlands Doodad
 98=Badlands Doodad
 99=Badlands Doodad
 100=Badlands Doodad
 101=Badlands Doodad
 102=Badlands Doodad
 103=Badlands Doodad
 104=Badlands Doodad
 105=Badlands Doodad
 106=Badlands Doodad
 107=Badlands Doodad
 108=Badlands Doodad
 109=Badlands Doodad
 110=Badlands Doodad
 111=Unknown 0
 112=Unknown 1
 113=Unknown 2
 114=Unknown 3
 115=Unknown 4
 116=Unknown 5
 117=Unknown 6
 118=Unknown 7
 119=Unknown 8
 120=Unknown 9
 121=Unknown 10
 122=Unknown 11
 123=Unknown 12
 124=Unknown 13
 125=Unknown 14
 126=Unknown 15
 127=Unknown 16
 128=Unknown 17
 129=Unknown 18
 130=Scourge
 131=Scourge Death
 132=Scourge Explosion
 133=Broodling

134=Broodling Death
 135=Infested Terran
 136=Infested Terran Explosion
 137=Guardian Cocoon
 138=Defiler
 139=Defiler Death
 140=Drone
 141=Drone Death
 142=Zerg Egg
 143=Zerg Egg Death
 144=Guardian
 145=Guardian Death
 146=Hydralisk
 147=Hydralisk Death
 148=Infested Kerrigan
 149=Larva
 150=Larva Death
 151=Mutalisk
 152=Mutalisk Death
 153=Overlord
 154=Overlord Death
 155=Queen
 156=Queen Birth
 157=Ultralisk
 158=Ultralisk Death
 159=Zergling
 160=Zergling Death
 161=Cerebrate
 162=Infested Command Center
 163=Spawning Pool
 164=Mature Chysalis
 165=Evolution Chamber
 166=Creep Colony
 167=Hatchery
 168=Hive
 169=Lair
 170=Sunken Colony
 171=Greater Spire
 172=Defiler Mound
 173=Queen's Nest
 174=Nydus Canal
 175=Overmind w/shell
 176=Overmind w/out shell
 177=Ultralisk Cavern
 178=Extractor
 179=Hydralisk Den
 180=Spire
 181=Spore Colony
 182=Zerg Building Spawn (Small)
 183=Zerg Building Spawn (Medium)
 184=Zerg Building Spawn (Large)
 185=Zerg Building Death
 186=Zerg Building Rubble (Large)
 187=Zerg Building Rubble (Small)
 188=Arbiter
 189=Archon Energy
 190=Carrier
 191=Dragoon
 192=Dragoon Death
 193=Interceptor
 194=Probe
 195=Scout
 196=Shuttle
 197=High Templar
 198=Dark Templar (Hero)
 199=Reaver
 200=Scarab

201=Zealot	276=Ragnasaur (Ashworld)	354=Unknown 79
202=Observer	277=Rynadon (Badlands)	355=Unknown 80
203=Templar Archives	278=Bengalass (Jungle)	356=Unknown 81
204=Assimilator	279=Mineral Deposit 1	357=Unknown 82
205=Observatory	280=Mineral Deposit 2	358=Unknown 83
206=Citadel of Adun	281=Mineral Deposit 3	359=Unknown 84
207=Forge	282=Unused	360=Unknown 85
208=Gateway	283=Zerg Beacon	361=Unknown 86
209=Cybernetics Core	284=Terran Beacon	362=Unknown 87
210=Khaydarin Crystal	285=Protoss Beacon	363=Consume
211=Nexus	286=Dark Swarm	364=Ensnare
212=Photon Cannon	287=Flag	365=Unknown 88
213=Arbiter Tribunal	288=Chrysalis	366=Unknown 89
214=Pylon	289=Psi Emmitter	367=Unknown 90
215=Robotics Facility	290=Data Disc	368=Unknown 91
216=Shield Battery	291=Khadarin Crystal	369=Phase Disruptor
217=Stargate	292=Mineral Chunk	370=Unknown 92
218=Stasis Cell/Prison	293=Unknown 28	371=Unknown 93
219=Robotics Support Bay	294=Protoss Gas Orb	372=Unknown 94
220=Protoss Temple	295=Unknown 29	373=Unknown 95
221=Fleet Beacon	296=Zerg Gas Sac	374=Unknown 96
222=Protoss Building Explosion (Large)	297=Unknown 30	375=Unknown 97
223=Protoss Building Rubble (Small)	298=Terran Gas Tank	376=Unknown 98
224=Protoss Building Rubble (Large)	299=Unknown 31	377=Bunker Overlay
225=Battlecruiser	300=Unknown 32	378=Unknown 99
226=Civilian	301=Start Location	379=Recall
227=Dropship	302=Unknown 33	380=Unknown 100
228=Firebat	303=Unknown 34	381=Unknown 101
229=Ghost	304=Unknown 35	382=Unknown 102
230=Ghost Death	305=Unknown 36	383=Unknown 103
231=Nuke Target	306=Unknown 37	384=Unknown 104
232=Goliath Base	307=Unknown 38	385=Unknown 105
233=Goliath Turret	308=Unknown 39	386=Unknown 106
234=Kerrigan (Ghost)	309=Unknown 40	387=Unknown 107
235=Marine	310=Unknown 41	388=Unknown 108
236=Marine Death	311=Unknown 42	389=Unknown 109
237=Unknown 19	312=Unknown 43	390=Unknown 110
238=Wraith	313=Unknown 44	391=Unknown 111
239=SCV	314=Unknown 45	392=Unknown 112
240=Siege Tank (Tank) Base	315=Unknown 46	393=Unknown 113
241=Siege Tank (Tank) Turret	316=Unknown 47	394=Unknown 114
242=Siege Tank (Siege) Base	317=Unknown 48	395=Unknown 115
243=Siege Tank (Siege) Turret	318=Unknown 49	396=Unknown 116
244=Vulture	319=Zerg Egg Spawn	397=Unknown 117
245=Spider Mine	320=Unknown 50	398=Unknown 118
246=Science Vessel	321=Unknown 51	399=Unknown 119
247=Science Vessel Turret	322=Unknown 52	400=Unknown 120
248=Academy	323=Unknown 53	401=Unknown 121
249=Barracks	324=Unknown 54	402=Unknown 122
250=Armory	325=Unknown 55	403=Unknown 123
251=Comsat Station	326=Unknown 56	404=Unknown 124
252=Command Center	327=Unknown 57	405=Unknown 125
253=Supply Depot	328=Unknown 58	406=Unknown 126
254=Control Tower	329=Unknown 59	407=Unknown 127
255=Factory	330=Unknown 60	408=Unknown 128
256=Covert Ops	331=Unknown 61	409=Unknown 129
257=Ion Cannon	332=Needle Spines	410=Unknown 130
258=Machine Shop	333=Unknown 62	411=Unknown 131
259=Missile Turret (Base)	334=Unknown 63	412=Unknown 132
260=Crashed Norad II	335=Unknown 64	413=Unknown 133
261=Physics Lab	336=Unknown 65	414=Unknown 134
262=Bunker	337=Phase Disruptor	415=Unknown 135
263=Refinery	338=Unknown 66	416=Unknown 136
264=Science Facility	339=Psi Storm	417=Unknown 137
265=Nuke Silo	340=Fusion Cutter	418=Unknown 138
266=Unknown 20	341=Unknown 67	419=Unknown 139
267=Unknown 21	342=Gemini Missles	420=Unknown 140
268=Starport	343=Unknown 68	421=Unknown 141
269=Unknown 22	344=Unknown 69	422=Unknown 142
270=Unknown 23	345=Unknown 70	423=Unknown 143
271=Unknown 24	346=Unknown 71	424=Unknown 144
272=Unknown 25	347=Unknown 72	425=Unknown 145
273=Unknown 26	348=Unknown 73	426=Unknown 146
274=Unknown 27	349=Unknown 74	427=Unknown 147
275=Vespene Geyser	350=Unknown 75	428=Unknown 148
	351=Unknown 76	429=Unknown 149
	352=Unknown 77	430=Unknown 150
	353=Unknown 78	431=Unknown 151

432=Unknown 152
 433=Unknown 153
 434=Unknown 154
 435=Unknown 155
 436=Unknown 156
 437=Unknown 157
 438=Unknown 158
 439=Unknown 159
 440=Unknown 160
 441=Unknown 161
 442=Unknown 162
 443=Unknown 163
 444=Unknown 164
 445=Unknown 165
 446=Unknown 166
 447=Unknown 167
 448=Unknown 168
 449=Unknown 169
 450=Unknown 170
 451=Unknown 171
 452=Unknown 172
 453=Unknown 173
 454=Unknown 174
 455=Unknown 175
 456=Unknown 176
 457=Unknown 177
 458=Unknown 178
 459=Unknown 179
 460=Unknown 180

461=Unknown 181
 462=Unknown 182
 463=Unknown 183
 464=Unknown 184
 465=Unknown 185
 466=Unknown 186
 467=Unknown 187
 468=Unknown 188
 469=Unknown 189
 470=Unknown 190
 471=Unknown 191
 472=Unknown 192
 473=Unknown 193
 474=Unknown 194
 475=Unknown 195
 476=Unknown 196
 477=Unknown 197
 478=Unknown 198
 479=Unknown 199
 480=Unknown 200
 481=Unknown 201
 482=Unknown 202
 483=Unknown 203
 484=Unknown 204
 485=Unknown 205
 486=Unknown 206
 487=Unknown 207
 488=Unknown 208
 489=Unknown 209

490=Unknown 210
 491=Unknown 211
 492=Unknown 212
 493=Unknown 213
 494=Unknown 214
 495=Unknown 215
 496=Unknown 216
 497=Unknown 217
 498=Unknown 218
 499=Unknown 219
 500=Unknown 220
 501=Unknown 221
 502=Unknown 222
 503=Unknown 223
 504=Unknown 224
 505=Unknown 225
 506=Unknown 226
 507=Unknown 227
 508=Unknown 228
 509=Unknown 229
 510=Unknown 230
 511=Unknown 231
 512=Unknown 232
 513=Unknown 233
 514=Unknown 234
 515=Unknown 235
 516=Unknown 236

Images.DAT:

000=Scourge
 001=Scourge Shad
 002=Scourge Birth
 003=Scourge Death
 004=Scourge Explosion
 005=Broodling
 006=Broodling Shad
 007=Broodling Death
 008=Infested Terran
 009=Infested Terran Shad
 010=Infested Terran Explosion
 011=Guardian Cocoon
 012=Guardian Cocoon Shad
 013=Defiler
 014=Defiler Shad
 015=Defiler Birth
 016=Defiler Death
 017=Drone
 018=Drone Shad
 019=Drone Birth
 020=Drone Death
 021=Zerg Egg
 022=Zerg Egg Shad
 023=Zerg Egg Spawn
 024=Zerg Egg Death
 025=Guardian
 026=Guardian Shad
 027=Guardian Birth
 028=Guardian Death
 029=Hydralisk
 030=Hydralisk Shad
 031=Hydralisk Birth
 032=Hydralisk Death
 033=Infested Kerrigan
 034=Infested Kerrigan Shad
 035=Needle Spines
 036=Larva
 037=Larva Death
 038=Mutalisk
 039=Mutalisk Shad
 040=Mutalisk Birth
 041=Mutalisk Death
 042=Overlord
 043=Overlord Shad

044=Overlord Birth
 045=Overlord Death
 046=Queen
 047=Queen Shad
 048=Queen Birth
 049=Queen Death
 050=Ultralisk
 051=Ultralisk Shad
 052=Ultralisk Birth
 053=Ultralisk Death
 054=Zergling
 055=Zergling Shad
 056=Zergling Birth
 057=Zergling Death
 058=Zerg Air Death (Large)
 059=Zerg Air Death (Small)
 060=Zerg Building Death
 061=Cerebrate
 062=Cerebrate Shad
 063=Infested Command Center
 064=Spawning Pool
 065=Spawning Pool Shad
 066=Evolution Chamber
 067=Evolution Chamber Shad
 068=Creep Colony
 069=Creep Colony Shad
 070=Hatchery
 071=Hatchery Shad
 072=Hive
 073=Hive Shad
 074=Lair
 075=Lair Shad
 076=Sunken Colony
 077=Sunken Colony Shad
 078=Mature Chysalis
 079=Mature Chysalis Shad
 080=Greater Spire
 081=Greater Spire Shad
 082=Defiler Mound
 083=Defiler Mound Shad
 084=Queen's Nest
 085=Queen's Nest Shad
 086=Nydus Canal
 087=Nydus Canal Shad

088=Overmind w/shell
 089=Overmind Shad
 090=Overmind w/out shell
 091=Ultralisk Cavern
 092=Ultralisk Cavern Shad
 093=Extractor
 094=Extractor Shad
 095=Hydralisk Den
 096=Hydralisk Den Shad
 097=Spire
 098=Spire Shad
 099=Spore Colony
 100=Spore Colony Shad
 101=Infested Command Center
 Overlay
 102=Zerg Building Morph 1
 103=Zerg Building Morph 2
 104=Zerg Building Morph 3
 105=Zerg Building Morph 4
 106=Zerg Building Morph Shad
 107=Zerg Building Spawn (Small)
 108=Zerg Building Spawn (Medium)
 109=Zerg Building Spawn (Large)
 110=Zerg Building Rubble (Large)
 111=Zerg Building Rubble (Small)
 112=Carrier
 113=Carrier Shad
 114=Carrier Glow
 115=Unknown
 116=Interceptor
 117=Interceptor Shad
 118=Shuttle
 119=Shuttle Shad
 120=Shuttle Glow
 121=Unknown
 122=Dragoon
 123=Dragoon Shad
 124=Dragoon Death
 125=Unknown
 126=High Templar
 127=High Templar Shad
 128=Unknown
 129=Dark Templar (Hero)
 130=Arbiter

131=Arbiter Shad	208=Fleet Beacon	281=Control Tower
132=Arbiter Glow	209=Unknown	282=Control Tower Attachment
133=Unknown	210=Protoss buildings warp texture	283=Control Tower Overlay
134=Archon Energy	211=Fleet Beacon Glow	284=Control Tower Shad
135=Archon Being	212=Fleet Beacon Shad	285=Factory
136=Archon Team Colors	213=Protoss Building Explosion (Small)	286=Factory Overlay
137=Probe	214=Protoss Building Explosion (Medium)	287=Factory Shad
138=Probe Shad	215=Protoss Building Explosion (Large)	288=Covert Ops
139=Unknown	216=Protoss Building Rubble (Small)	289=Covert Ops Attachment
140=Scout	217=Protoss Building Rubble (Large)	290=Covert Ops Overlay
141=Scout Shad	218=Battlecruiser	291=Covert Ops Shad
142=Scout Glow	219=Battlecruiser Shad	292=Ion Cannon
143=Unknown	220=Battlecruiser Glow	293=Machine Shop
144=Reaver	221=Civilian	294=Machine Shop Attachment
145=Reaver Shad	222=Civilian Shad	295=Machine Shop Shad
146=Unknown	223=Dropship	296=Missile Turret (Base)
147=Scarab	224=Dropship Shad	297=Missile Turret (Turret)
148=Observer	225=Dropship Glow	298=Missile Turret Shad
149=Observer Shad	226=Firebat	299=Crashed Norad II
150=Unknown	227=Firebat Shad	300=Crashed Norad II Shad
151=Zealot	228=Ghost	301=Physics Lab
152=Zealot Shad	229=Ghost Shad	302=Physics Lab Attachment
153=Unknown	230=Ghost Death	303=Physics Lab Shad
154=Unknown	231=Unknown	304=Bunker
155=Templar Archives	232=Nuke Beam	305=Bunker Shad
156=Unknown	233=Nuke Target	306=Bunker Overlay
157=Templar Archives Shad	234=Goliath Base	307=Refinery
158=Assimilator	235=Goliath Turret	308=Refinery Shad
159=Unknown	236=Goliath Shad	309=Science Facility
160=Unknown	237=Kerrigan (Ghost)	310=Science Facility Overlay
161=Observatory	238=Kerrigan (Ghost) Shad	311=Science Facility Shad
162=Observatory Warp Flash*	239=Marine	312=Nuke Silo
163=Observatory Shad	240=Marine Shad	313=Nuke Silo Attachment
164=Citadel of Adun	241=Marine Death	314=Nuke Silo Overlay
165=Citadel of Adun Warp Flash*	242=Unknown	315=Nuke Silo Shad
166=Citadel of Adun Shad	243=Wraith	316=Nuclear Missile
167=Forge	244=Wraith Shad	317=Nuclear Missile Shad
168=Forge Overlay	245=Wraith Glow	318=Nuclear Explosion
169=Forge Warp Flash*	246=Unknown	319=Starport
170=Forge Shad	247=SCV	320=Starport Overlay
171=Gateway	248=SCV Shad	321=Starport Shad
172=Gateway Warp Flash*	249=SCV Glow	322=Engineering Bay
173=Gateway Shad	250=Siege Tank (Tank) Base	323=Engineering Bay Overlay
174=Cybernetics Core	251=Siege Tank (Tank) Turret	324=Engineering Bay Shad
175=Cybernetics Core Warp Flash*	252=Siege Tank (Tank) Shad	325=Construction Site (large)
176=Cybernetics Core Overlay	253=Siege Tank (Siege) Base	326=Construction Site (large) Shad
177=Cybernetics Core Shad	254=Siege Tank (Siege) Turret	327=Construction Site (medium)
178=Khaydarin Crystal	255=Siege Tank (Siege) Shad	328=Construction Site (medium) Shad
179=Nexus	256=Vulture	329=Construction Site (small)
180=Nexus Warp Flash*	257=Vulture Shad	330=Construction Site (small) addon
181=Nexus Glow	258=Spider Mine	331=Construction Site (small) Shad
182=Nexus Shad	259=Spider Mine Shad	332=Terran Building Explosion (small)
183=Photon Cannon	260=Science Vessel	333=Terran Building Explosion (medium)
184=Photon Cannon Shad	261=Science Vessel Turret	334=Terran Building Explosion (large)
185=Photon Cannon Warp Flash*	262=Science Vessel Shad	335=Terran Building Rubble (small)
186=Arbiter Tribunal	263=Academy	336=Terran Building Rubble (large)
187=Arbiter Tribunal Warp Flash*	264=Academy Overlay	337=Dark Swarm
188=Arbiter Tribunal Shad	265=Academy Shad	338=Ragnasaur (Ashworld)
189=Pylon	266=Barracks	339=Ragnasaur Shad
190=Pylon Warp Flash*	267=Barracks Shad	340=Rynadon (Badlands)
191=Pylon Shad	268=Armory	341=Rynadon Shad
192=Robotics Facility	269=Armory Overlay	342=Bengalass (Jungle)
193=Robotics Facility Warp Flash*	270=Armory Shad	343=Bengalass Shad
194=Robotics Facility Shad	271=Comsat Station	344=Vespene Geyser
195=Shield Battery	272=Comsat Station Attachment	345=Unknown
196=Shield Battery Glow	273=Comsat Station Overlay	346=Vespene Geyser Shad
197=Shield Battery Warp Flash*	274=Comsat Station Shad	347=Mineral Deposit 1
198=Shield Battery Shad	275=Command Center	348=Mineral Deposit 1 Shad
199=Stargate	276=Command Center Overlay	349=Mineral Deposit 2
200=Stargate Glow	277=Command Center Shad	350=Mineral Deposit 2 Shad
201=Stargate Warp Flash*	278=Supply Depot	351=Mineral Deposit 3
202=Stargate Shad	279=Supply Depot Overlay	352=Mineral Deposit 3 Shad
203=Stasis Cell/Prison	280=Supply Depot Shad	353=Unused(scout copy)
204=Robotics Support Bay		
205=Robotics Support Bay Warp Flash*		
206=Robotics Support Bay Shad		
207=Protoss Temple		

354=Zerg Beacon	425=Unknown	503=Unknown
355=Zerg Beacon Overlay	426=Unknown	504=Unknown
356=Terran Beacon	427=Unknown	505=Unknown
357=Terran Beacon Overlay	428=Small explosion	506=Unknown
358=Protoss Beacon	429=Spider mine explosion	507=Unknown
359=Protoss Beacon Overlay	430=Unknown	508=Unknown
360=lockdown missile (unused) - white beam	431=Unknown	509=Unknown
361=lockdown (unit overlay) small	432=Unknown	510=Unknown
362=lockdown (unit overlay) medium	433=Unknown	511=Unknown
363=lockdown (unit overlay) large	434=Unknown	512=Unknown
364=stasis field hit	435=Unknown	513=Unknown
365=stasis field (unit overlay) small	436=Unknown	514=Unknown
366=stasis field (unit overlay) medium	437=Unknown	515=Unknown
367=stasis field (unit overlay) large	438=Unknown	516=Parasite
368=shield battery (unit overlay) small	439=Unknown	517=Consume
369=shield battery (unit overlay) medium	440=Fragmentation grenade hit	518=Unknown
370=shield battery (unit overlay) large	441=Fragmentation grenade smoke	519=Unknown
371=defence matrix front (small)	442=Unknown	520=Unknown
372=defence matrix front (medium)	443=Scarab Trail	521=Unknown
373=defence matrix front (large)	444=Scarab Explosion	522=Photon Cannons (Interceptor)
374=defence matrix back (small)	445=Unknown	523=Phase Disruptor
375=defence matrix back (medium)	446=BC Laser Fire Overlay	524=Unknown
376=defence matrix back (large)	447=Laser Hit	525=Psi Storm
377=defence matrix hit (small)	448=Unknown	526=Fusion Cutter
378=defence matrix hit (medium)	449=High Templar Trail	527=Unknown
379=defence matrix hit (large)	450=Terran Building Burn	528=Gemini Missiles
380=Irradiate (small)	451=Terran Building Burn	529=Unknown
381=Irradiate (medium)	452=Terran Building Burn	530=Unknown
382=Irradiate (large)	453=Terran Building Burn	531=Unknown
383=Ensnare	454=Terran Building Burn	532=Unknown
384=Ensnare (Unit Overlay) Small	455=Terran Building Burn	533=Unknown
385=Ensnare (Unit Overlay) Medium	456=Terran Building Burn	534=Unknown
386=Ensnare (Unit Overlay) Large	457=Terran Building Burn	535=Unknown
387=Plague	458=Zerg Building Blood	536=Unknown
388=Plague (Unit Overlay) Small	459=Zerg Building Blood	537=Unknown
389=Plague (Unit Overlay) Medium	460=Zerg Building Blood	538=Unknown
390=Plague (Unit Overlay) Large	461=Zerg Building Blood	539=Unknown
391=Recall	462=Zerg Building Blood	540=Unknown
392=Flag	463=Zerg Building Blood	541=Unknown
393=Chrysalis	464=Zerg Building Blood	542=Unknown
394=Psi Emmitter	465=Zerg Building Blood	543=Unknown
395=Data Disc	466=Protoss Building Burn	544=Unknown
396=Khadarin Crystal	467=Protoss Building Burn	545=Unknown
397=Mineral Chunk	468=Protoss Building Burn	546=Unknown
398=Unknown	469=Protoss Building Burn	547=Unknown
399=Protoss Gas Orb	470=Protoss Building Burn	548=Unknown
400=Unknown	471=Protoss Building Burn	549=Unknown
401=Zerg Gas Sac	472=Terran Building Burn	550=Unknown
402=Unknown	473=Terran Building Burn	551=Unknown
403=Terran Gas Tank	474=Terran Building Burn	552=Unknown
404=Unknown	475=Terran Building Burn	553=Unknown
405=Mineral Chunk Shad	476=Terran Building Burn	554=Unknown
406=Protoss Gas Orb Shad	477=Terran Building Burn	555=Unknown
407=Zerg Gas Sack Shad	478=Terran Building Burn	556=Unknown
408=Terran Gas Tank Shad	479=Terran Building Burn	557=Unknown
409=Data Disk Shad (Ground)	480=Zerg Building Blood	558=Unknown
410=Data Disk Shad (Carried)	481=Zerg Building Blood	559=Unknown
411=Flag Shad (Ground)	482=Zerg Building Blood	560=Unknown
412=Flag Shad (Carried)	483=Zerg Building Blood	561=o022
413=Crystal Shad (Ground)	484=Zerg Building Blood	562=o032
414=Crystal Shad (Carried)	485=Zerg Building Blood	563=o048
415=Chrysalis Shad (Ground)	486=Zerg Building Blood	564=o062
416=Chrysalis Shad (Carried)	487=Zerg Building Blood	565=o072
417=Psi Emitter Shad (Ground)	488=Protoss Building Burn	566=o094
418=Psi Emitter Shad (Carried)	489=Protoss Building Burn	567=o110
419=Unknown	490=Protoss Building Burn	568=o122
420=Unknown	491=Protoss Building Burn	569=o146
421=Flame Thrower	492=Protoss Building Burn	570=o224
422=Missile Trail	493=Protoss Building Burn	571=od022
423=Unknown	494=Protoss Building Burn	572=od032
424=Plasma Shields	495=Unknown	573=od048
	496=Unknown	574=od062
	497=Unknown	575=od072
	498=Unknown	576=od094
	499=Unknown	577=od110
	500=Unknown	578=od122
	501=Unknown	579=od146
	502=Unknown	580=od224

581=Unknown	659=Jungle Doodad	737=Installation Doodad
582=Map Revealer	660=Jungle Doodad	738=Installation Doodad
583=Unknown	661=Jungle Doodad	739=Installation Doodad
584=Psi Field 1	662=Jungle Doodad	740=Installation Doodad
585=Psi Field 2	663=Jungle Doodad	741=Installation Doodad
586=Psi Field 3	664=Jungle Doodad	742=Installation Doodad
587=Psi Field 4	665=Jungle Doodad	743=Installation Doodad
588=Start Location	666=Space Platform Doodad	744=Installation Doodad
589=Ash World Doodad (Rock 1)	667=Space Platform Doodad	745=Installation Doodad
590=Ash World Doodad (Rock 1 Shad)	668=Space Platform Doodad	746=Installation Doodad
591=Ash World Doodad (Rock 2)	669=Space Platform Doodad	747=Installation Doodad
592=Ash World Doodad (Rock 2 Shad)	670=Space Platform Doodad	748=Installation Doodad
593=Ash World Doodad (Rock 3)	671=Space Platform Doodad	749=Installation Doodad
594=Ash World Doodad (Rock 3 Shad)	672=Space Platform Doodad	750=Installation Doodad
595=Ash World Doodad (Rock 4)	673=Space Platform Doodad	751=Installation Doodad
596=Ash World Doodad (Rock 4 Shad)	674=Space Platform Doodad	752=Installation Doodad
597=Ash World Doodad (Rock 5)	675=Space Platform Doodad	753=Installation Doodad
598=Ash World Doodad (Rock 5 Shad)	676=Space Platform Doodad	754=Installation Doodad
599=Ash World Doodad	677=Space Platform Doodad	755=Twilight Doodad
600=Ash World Doodad	678=Space Platform Doodad	756=Twilight Doodad
601=Ash World Doodad	679=Space Platform Doodad	757=Twilight Doodad
602=Ash World Doodad	680=Space Platform Doodad	758=Twilight Doodad
603=Ash World Doodad	681=Space Platform Doodad	759=Twilight Doodad
604=Ash World Doodad	682=Space Platform Doodad	760=Twilight Doodad
605=Ash World Doodad	683=Space Platform Doodad	761=Twilight Doodad
606=Ash World Doodad	684=Space Platform Doodad	762=Twilight Doodad
607=Ash World Doodad	685=Space Platform Doodad	763=Twilight Doodad
608=Ash World Doodad	686=Space Platform Doodad	764=Twilight Doodad
609=Ash World Doodad	687=Space Platform Doodad	765=Twilight Doodad
610=Jungle Doodad	688=Space Platform Doodad	766=Twilight Doodad
611=Jungle Doodad	689=Space Platform Doodad	767=Twilight Doodad
612=Jungle Doodad	690=Space Platform Doodad	768=Twilight Doodad
613=Jungle Doodad	691=Badlands Doodad	769=Twilight Doodad
614=Jungle Doodad	692=Badlands Doodad	770=Twilight Doodad
615=Jungle Doodad	693=Badlands Doodad	771=Ice World Doodad
616=Jungle Doodad	694=Badlands Doodad	772=Ice World Doodad
617=Jungle Doodad	695=Badlands Doodad	773=Ice World Doodad
618=Jungle Doodad	696=Badlands Doodad	774=Ice World Doodad
619=Jungle Doodad	697=Badlands Doodad	775=Ice World Doodad
620=Jungle Doodad	698=Badlands Doodad	776=Ice World Doodad
621=Jungle Doodad	699=Badlands Doodad	777=Ice World Doodad
622=Jungle Doodad	700=Badlands Doodad	778=Ice World Doodad
623=Jungle Doodad	701=Badlands Doodad	779=Ice World Doodad
624=Jungle Doodad	702=Badlands Doodad	780=Ice World Doodad
625=Jungle Doodad	703=Badlands Doodad	781=Ice World Doodad
626=Jungle Doodad	704=Badlands Doodad	782=Ice World Doodad
627=Jungle Doodad	705=Badlands Doodad	783=Ice World Doodad
628=Jungle Doodad	706=Badlands Doodad	784=Ice World Doodad
629=Jungle Doodad	707=Badlands Doodad	785=Ice World Doodad
630=Jungle Doodad	708=Badlands Doodad	786=Ice World Doodad
631=Jungle Doodad	709=Badlands Doodad	787=Ice World Doodad
632=Jungle Doodad	710=Badlands Doodad	788=Ice World Doodad
633=Jungle Doodad	711=Badlands Doodad	789=Ice World Doodad
634=Jungle Doodad	712=Badlands Doodad	790=Ice World Doodad
635=Jungle Doodad	713=Badlands Doodad	791=Ice World Doodad
636=Jungle Doodad	714=Badlands Doodad	792=Ice World Doodad
637=Jungle Doodad	715=Badlands Doodad	793=Ice World Doodad
638=Jungle Doodad	716=Badlands Doodad	794=Ice World Doodad
639=Jungle Doodad	717=Badlands Doodad	795=Ice World Doodad
640=Jungle Doodad	718=Badlands Doodad	796=Ice World Doodad
641=Jungle Doodad	719=Badlands Doodad	797=Ice World Doodad
642=Jungle Doodad	720=Badlands Doodad	798=Ice World Doodad
643=Jungle Doodad	721=Badlands Doodad	799=Ice World Doodad
644=Jungle Doodad	722=Badlands Doodad	800=Ice World Doodad
645=Jungle Doodad	723=Badlands Doodad	801=Ice World Doodad
646=Jungle Doodad	724=Badlands Doodad	802=Ice World Doodad
647=Jungle Doodad	725=Badlands Doodad	803=Ice World Doodad
648=Jungle Doodad	726=Badlands Doodad	804=Ice World Doodad
649=Jungle Doodad	727=Badlands Doodad	805=Ice World Doodad
650=Jungle Doodad	728=Badlands Doodad	806=Ice World Doodad
651=Jungle Doodad	729=Badlands Doodad	807=Ice World Doodad
652=Jungle Doodad	730=Badlands Doodad	808=Ice World Doodad
653=Jungle Doodad	731=Badlands Doodad	809=Ice World Doodad
654=Jungle Doodad	732=Badlands Doodad	810=Ice World Doodad
655=Jungle Doodad	733=Badlands Doodad	811=Ice World Doodad
656=Jungle Doodad	734=Badlands Doodad	812=Ice World Doodad
657=Jungle Doodad	735=Installation Doodad	813=Ice World Doodad
658=Jungle Doodad	736=Installation Doodad	814=Ice World Doodad

815=Ice World Doodad	877=Desert Doodad	939=Valkyrie
816=Ice World Doodad	878=Desert Doodad	940=Valkyrie Shad
817=Ice World Doodad	879=Desert Doodad	941=Valkyrie Overlay
818=Ice World Doodad	880=Desert Doodad	942=Valkyrie Overlay
819=Ice World Doodad	881=Desert Doodad	943=Valkyrie Afterburners
820=Ice World Doodad	882=Desert Doodad	944=Medic
821=Ice World Doodad	883=Desert Doodad	945=Medic Shad
822=Ice World Doodad	884=Desert Doodad	946=Medic Death
823=Ice World Doodad	885=Desert Doodad	947=Psi Disruptor
824=Ice World Doodad	886=Desert Doodad	948=Psi Disruptor Shad
825=Ice World Doodad	887=Desert Doodad	949=Power Generator
826=Ice World Doodad	888=Desert Doodad	950=Power Generator Shad
827=Ice World Doodad	889=Desert Doodad	951=Disruption Web
828=Ice World Doodad	890=Desert Doodad	952=Scantid
829=Ice World Doodad	891=Desert Doodad	953=Scantid Shad
830=Ice World Doodad	892=Desert Doodad	954=Kakaru
831=Ice World Doodad	893=Desert Doodad	955=Kakaru Shad
832=Ice World Doodad	894=Desert Doodad	956=Ursadon
833=Ice World Doodad	895=Desert Doodad	957=Ursadon Shad
834=Ice World Doodad	896=Desert Doodad	958=Uraj Crystal
835=Ice World Doodad	897=Desert Doodad	959=Khalis Crystal
836=Ice World Doodad	898=Desert Doodad	960=Unknown
837=Ice World Doodad	899=Desert Doodad	961=Subterranean Spines
838=Ice World Doodad	900=Desert Doodad	962=Acid Spores Projectile
839=Ice World Doodad	901=Desert Doodad	963=Acid Spores Hit
840=Ice World Doodad	902=Desert Doodad	964=Neutron Flare
841=Ice World Doodad	903=Desert Doodad	965=Halo Rocket
842=Ice World Doodad	904=Desert Doodad	966=Optic Flare Projectile
843=Ice World Doodad	905=Desert Doodad	967=Restoration (Small)
844=Ice World Doodad	906=Desert Doodad	968=Restoration (Medium)
845=Ice World Doodad	907=Desert Doodad	969=Restoration (Large)
846=Ice World Doodad	908=Desert Doodad	970=Unknown
847=Ice World Doodad	909=Desert Doodad	971=Unknown
848=Ice World Doodad	910=Desert Doodad	972=Unknown
849=Ice World Doodad	911=Desert Doodad	973=Mind Control (Small)
850=Ice World Doodad	912=Desert Doodad	974=Mind Control (Medium)
851=Desert Doodad	913=Unused (Scout)	975=Mind Control (Large)
852=Desert Doodad	914=Lurker Egg	976=Optic Flare (Small)
853=Desert Doodad	915=Devourer	977=Optic Flare (Medium)
854=Desert Doodad	916=Devourer Shad	978=Optic Flare (Large)
855=Desert Doodad	917=Devourer Birth	979=Feedback (Small)
856=Desert Doodad	918=Devourer Death	980=Feedback (Medium)
857=Desert Doodad	919=Lurker Birth	981=Feedback (Large)
858=Desert Doodad	920=Unknown	982=Maelstrom Unit Overlay (Small)
859=Desert Doodad	921=Unknown	983=Maelstrom Unit Overlay (Medium)
860=Desert Doodad	922=Lurker Shad	984=Maelstrom Unit Overlay (Large)
861=Desert Doodad	923=Overmind Cocoon	985=Unknown
862=Desert Doodad	924=Overmind Cocoon Shad	986=Acid Spores Overlay 1 (Small)
863=Desert Doodad	925=Dark Archon Energy	987=Acid Spores Overlay 2 (Small)
864=Desert Doodad	926=Dark Archon Being	988=Acid Spores Overlay 3 (Small)
865=Desert Doodad	927=Dark Archon Team Colors	989=Acid Spores Overlay 4 (Small)
866=Desert Doodad	928=Dark Archon Death	990=Acid Spores Overlay 1 (Medium)
867=Desert Doodad	929=Corsair	991=Acid Spores Overlay 2 (Medium)
868=Desert Doodad	930=Corsair Shad	992=Acid Spores Overlay 3 (Medium)
869=Desert Doodad	931=Corsair Overlay	993=Acid Spores Overlay 4 (Medium)
870=Desert Doodad	932=Unknown	994=Acid Spores Overlay 1 (Large)
871=Desert Doodad	933=Dark Templar (BW Unit)	995=Acid Spores Overlay 2 (Large)
872=Desert Doodad	934=Warp Gate	996=Acid Spores Overlay 3 (Large)
873=Desert Doodad	935=Warp Gate Shad	997=Acid Spores Overlay 4 (Large)
874=Desert Doodad	936=Warp Gate Overlay	998=Maelstrom (Hit)
875=Desert Doodad	937=Xel'Naga Temple	
876=Desert Doodad	938=Xel'Naga Temple Shad	

Sfxdata.DAT:

(Note that in the iscript the sound effects numbering starts from 1, not 0 like in Arsenal, so if you use Arsenal to determine sound numbers, use the number + 1)

1=Zerg\Drone\ZDrErr00.WAV	10=Misc\Explo1.wav	19=Misc\IntoNyduus.wav
2=Misc\Buzz.wav	11=Misc\Explo2.wav	20=Misc\OutOfGas.wav
3=Misc\PErr0r.WAV	12=Misc\Explo3.wav	21=Misc\youwin.wav
4=Misc\ZBldgPlc.wav	13=Misc\Explo4.wav	22=Misc\youlose.wav
5=Misc\TBldgPlc.wav	14=Misc\Explo5.wav	23=Misc\Transmission.wav
6=Misc\PBldgPlc.wav	15=Misc\Button.wav	24=Misc\Door\Door1Opn.wav
7=Misc\ExploLrg.wav	16=Misc\BurrowDn.wav	25=Misc\Door\Door1Cls.wav
8=Misc\ExploMed.wav	17=Misc\BurrowUp.wav	26=Misc\Door\Door2Opn.wav
9=Misc\ExploSm.wav	18=Misc\OnFirSml.wav	27=Misc\Door\Door2Cls.wav

28=Misc\Door\Door3Opn.wav
 29=Misc\Door\Door3Cls.wav
 30=Misc\Door\Door4Opn.wav
 31=Misc\Door\Door4Cls.wav
 32=Misc\ZRescue.wav
 33=Misc\TRescue.wav
 34=Misc\PRescue.wav
 35=terran\SCV\EDrRep00.wav
 36=terran\SCV\EDrRep01.wav
 37=terran\SCV\EDrRep02.wav
 38=terran\SCV\EDrRep03.wav
 39=terran\SCV\EDrRep04.wav
 40=Misc\ZOvTra00.wav
 41=Misc\TDtra00.wav
 42=Misc\PShTra00.wav
 43=Misc\ZOvTra01.wav
 44=Misc\TDtra01.wav
 45=Misc\PShTra01.wav
 46=Misc\CRITTERS\JCrWh00.wav
 47=Misc\CRITTERS\JCrWh01.wav
 48=Misc\CRITTERS\JCrWh02.wav
 49=Misc\CRITTERS\JCrDth00.wav
 50=Misc\CRITTERS\LcrWh00.wav
 51=Misc\CRITTERS\LcrWh01.wav
 52=Misc\CRITTERS\LcrWh02.wav
 53=Misc\CRITTERS\LcrDth00.wav
 54=Misc\CRITTERS\BcrWh00.wav
 55=Misc\CRITTERS\BcrWh01.wav
 56=Misc\CRITTERS\BcrWh02.wav
 57=Misc\CRITTERS\BcrDth00.wav
 58=Bullet\ParFir00.wav
 59=Bullet\ZHyFir00.wav
 60=Bullet\ZHyHit00.wav
 61=Bullet\ZDrHit00.wav
 62=Bullet\LaserHit.wav
 63=Bullet\pshield.wav
 64=Bullet\SpooGHit.wav
 65=Bullet\BlastCan.wav
 66=Bullet\BlastGn2.wav
 67=Bullet\ZBGHit00.wav
 68=Bullet\TTaFir00.wav
 69=Bullet\TMaFir00.wav
 70=Bullet\TGoFir00.wav
 71=Bullet\ZGuFir00.wav
 72=Bullet\ZGuHit00.wav
 73=Bullet\LASERB.wav
 74=Bullet\TPhFi100.wav
 75=Bullet\LASRHIT1.wav
 76=Bullet\LASRHIT2.wav
 77=Bullet\LASRHIT3.wav
 78=Bullet\ZLuFir00.wav
 79=Bullet\ZLuHit00.wav
 80=Bullet\HKMISSLE.wav
 81=Bullet\TGoFi200.wav
 82=Bullet\TPhFi200.wav
 83=Bullet\TNsFir00.wav
 84=Bullet\TNsFir00.wav
 85=Bullet\TNsHit00.wav
 86=Bullet\PhoAtt00.wav
 87=Bullet\PhoHit00.wav
 88=Bullet\PSIBLADE.wav
 89=Bullet\PSIBOLT.wav
 90=Bullet\ZQuFir00.wav
 91=Bullet\ZQuHit00.wav
 92=Bullet\ZQuHit01.wav
 93=Bullet\ZQuHit02.wav
 94=Bullet\tsCFir00.wav
 95=Bullet\SHOCKBMB.wav
 96=Bullet\TTaFi200.wav
 97=Bullet\SHCKLNCH.wav
 98=Bullet\TGHFir00.wav
 99=Bullet\DragBull.wav
 100=Bullet\psaHit00.wav
 101=Bullet\LaserB.wav
 102=Bullet\pTrFir00.wav
 103=Bullet\pTrFir01.wav
 104=Bullet\pzeFir00.wav
 105=Bullet\tbaFir00.wav
 106=Bullet\tvuFir00.wav
 107=Bullet\tvuHit00.wav
 108=Bullet\tvuHit01.wav
 109=Bullet\tvuHit02.wav
 110=Bullet\zdeAtt00.wav
 111=Bullet\UZeFir00.wav
 112=Bullet\UKiFir00.wav
 113=Bullet\zmuFir00.wav
 114=Misc\ZPwrDown.wav
 115=Misc\TPwrDown.wav
 116=Misc\PPwrDown.wav
 117=Zerg\Advisor\ZAdUpd00.WAV
 118=Terran\Advisor\TAdUpd00.WAV
 119=Protoss\Advisor\PAAdUpd00.WAV
 120=Zerg\Advisor\ZAdUpd01.WAV
 121=Terran\Advisor\TAdUpd01.WAV
 122=Protoss\Advisor\PAAdUpd01.WAV
 123=Zerg\Advisor\ZAdUpd02.WAV
 124=Terran\Advisor\TAdUpd02.WAV
 125=Protoss\Advisor\PAAdUpd02.WAV
 126=Terran\Advisor\TAdUpd03.WAV
 127=Zerg\Advisor\ZAdUpd04.WAV
 128=Terran\Advisor\TAdUpd04.WAV
 129=Protoss\Advisor\PAAdUpd04.WAV
 130=Terran\Advisor\TAdUpd05.WAV
 131=Terran\Advisor\TAdUpd06.WAV
 132=Terran\Advisor\TAdUpd02.WAV
 133=Protoss\Advisor\PAAdUpd06.WAV
 134=Terran\Advisor\TAdUpd07.WAV
 135=Zerg\Bldg\ZChRdy00.WAV
 136=Terran\SCV\TSCUpd00.WAV
 137=Zerg\DRONE\ZDrErr00.WAV
 138=Zerg\DRONE\ZDrErr00.WAV
 139=Misc\Buzz.wav
 140=Misc\PErr00.WAV
 141=Misc\ZBldgPlc.wav
 142=Terran\Advisor\tAdErr04.WAV
 143=Terran\Advisor\tAdErr03.WAV
 144=Terran\Advisor\tAdErr03.WAV
 145=Terran\Advisor\tAdErr04.WAV
 146=Zerg\Advisor\ZAdErr00.WAV
 147=Zerg\Advisor\ZAdErr00.WAV
 148=Terran\Advisor\tAdErr00.WAV
 149=Protoss\Advisor\PAAdErr00.WAV
 150=Zerg\Advisor\ZAdErr01.WAV
 151=Terran\Advisor\tAdErr01.WAV
 152=Protoss\Advisor\PAAdErr01.WAV
 153=Zerg\Advisor\ZAdErr02.WAV
 154=Terran\Advisor\tAdErr02.WAV
 155=Protoss\Advisor\PAAdErr02.WAV
 156=Zerg\Advisor\ZAdErr06.WAV
 157=Terran\Advisor\tAdErr06.WAV
 158=Protoss\Advisor\PAAdErr06.WAV
 159=Terran\SCV\TSCErr01.WAV
 160=Terran\Advisor\tAdErr04.WAV
 161=Protoss\PROBE\PPPrErr00.WAV
 162=Terran\SCV\TSCErr00.WAV
 163=Terran\Advisor\tAdErr04.WAV
 164=Protoss\PROBE\PPPrErr01.WAV
 165=glue\mousedown2.wav
 166=glue\mouseover.wav
 167=glue\bnctclick.wav
 168=glue\scorefill.wav
 169=glue\scorefillend.wav
 170=glue\countdown.wav
 171=glue\swishlock.wav
 172=glue\swishin.wav
 173=glue\swishout.wav
 174=Terran\SCV\TSCMin00.wav
 175=Terran\SCV\TSCMin01.wav
 176=Terran\BATTLE\tbady00.wav
 177=Terran\BATTLE\tbaDth00.wav
 178=Terran\BATTLE\tBaYam01.wav
 179=Terran\BATTLE\tBaYam02.wav
 180=Terran\BATTLE\tbapss00.wav
 181=Terran\BATTLE\tbapss01.wav
 182=Terran\BATTLE\tbapss02.wav
 183=Terran\BATTLE\tbapss03.wav
 184=Terran\BATTLE\tbapss04.wav
 185=Terran\BATTLE\tbawht00.wav
 186=Terran\BATTLE\tbawht01.wav
 187=Terran\BATTLE\tbawht02.wav
 188=Terran\BATTLE\tbawht03.wav
 189=Terran\BATTLE\tbayes00.wav
 190=Terran\BATTLE\tbayes01.wav
 191=Terran\BATTLE\tbayes02.wav
 192=Terran\BATTLE\tbayes03.wav
 193=Terran\CIVILIAN\TCvPss00.WAV
 194=Terran\CIVILIAN\TCvPss01.WAV
 195=Terran\CIVILIAN\TCvPss02.WAV
 196=Terran\CIVILIAN\TCvPss03.WAV
 197=Terran\CIVILIAN\TCvPss04.WAV
 198=Terran\CIVILIAN\TCvWh00.WAV
 199=Terran\CIVILIAN\TCvWh01.WAV
 200=Terran\CIVILIAN\TCvWh02.WAV
 201=Terran\CIVILIAN\TCvWh03.WAV
 202=Terran\CIVILIAN\TCvYes00.WAV
 203=Terran\CIVILIAN\TCvYes01.WAV
 204=Terran\CIVILIAN\TCvYes02.WAV
 205=Terran\CIVILIAN\TCvYes03.WAV
 206=Terran\CIVILIAN\TCvYes04.WAV
 207=Terran\CIVILIAN\TCvDth00.WAV
 208=Terran\CIVILIAN\TCvRdy00.WAV
 209=Terran\DROPSHIP\TDrdy00.wav
 210=Terran\DROPSHIP\TDrdth00.wav
 211=Terran\DROPSHIP\TDrdss00.wav
 212=Terran\DROPSHIP\TDrdss01.wav
 213=Terran\DROPSHIP\TDrdss02.wav
 214=Terran\DROPSHIP\TDrdss03.wav
 215=Terran\DROPSHIP\TDrdWh00.wav
 216=Terran\DROPSHIP\TDrdWh01.wav
 217=Terran\DROPSHIP\TDrdWh02.wav
 218=Terran\DROPSHIP\TDrdWh03.wav
 219=Terran\DROPSHIP\TDrdYes00.wav
 220=Terran\DROPSHIP\TDrdYes01.wav
 221=Terran\DROPSHIP\TDrdYes02.wav
 222=Terran\DROPSHIP\TDrdYes03.wav
 223=Terran\DROPSHIP\TDrdYes04.wav
 224=Terran\DROPSHIP\TDrdYes05.wav
 225=Terran\GHOST\TGhRdy00.wav
 226=Terran\GHOST\TGhPss00.wav
 227=Terran\GHOST\TGhPss01.wav
 228=Terran\GHOST\TGhPss02.wav
 229=Terran\GHOST\TGhPss03.wav
 230=Terran\GHOST\TGhWh00.wav
 231=Terran\GHOST\TGhWh01.wav
 232=Terran\GHOST\TGhWh02.wav
 233=Terran\GHOST\TGhWh03.wav
 234=Terran\GHOST\TGhYes00.wav
 235=Terran\GHOST\TGhYes01.wav
 236=Terran\GHOST\TGhYes02.wav
 237=Terran\GHOST\TGhYes03.wav
 238=Terran\GHOST\TGhDth00.wav
 239=Terran\GHOST\TGhLas00.wav
 240=Terran\GHOST\TGhLkd00.wav
 241=Terran\GOLIATH\TGoRdy00.wav
 242=Terran\GOLIATH\TGoPss00.wav
 243=Terran\GOLIATH\TGoPss01.WAV
 244=Terran\GOLIATH\TGoPss02.WAV
 245=Terran\GOLIATH\TGoPss03.WAV
 246=Terran\GOLIATH\TGoPss04.WAV
 247=Terran\GOLIATH\TGoPss05.WAV
 248=Terran\GOLIATH\TGoWh00.WAV
 249=Terran\GOLIATH\TGoWh01.WAV
 250=Terran\GOLIATH\TGoWh02.WAV
 251=Terran\GOLIATH\TGoWh03.WAV
 252=Terran\GOLIATH\TGoYes00.WAV
 253=Terran\GOLIATH\TGoYes01.WAV
 254=Terran\GOLIATH\TGoYes02.WAV
 255=Terran\GOLIATH\TGoYes03.WAV
 256=Terran\PHOENIX\TPhRdy00.WAV
 257=Terran\PHOENIX\TPhDth00.WAV
 258=Terran\PHOENIX\TPhPss00.WAV
 259=Terran\PHOENIX\TPhPss01.WAV
 260=Terran\PHOENIX\TPhPss02.WAV
 261=Terran\PHOENIX\TPhPss03.WAV

262=Terran\PHOENIX\TPhPss04.WAV
 263=Terran\PHOENIX\TPhPss05.WAV
 264=Terran\PHOENIX\TPhPss06.WAV
 265=Terran\PHOENIX\TPhWht00.WAV
 266=Terran\PHOENIX\TPhWht01.WAV
 267=Terran\PHOENIX\TPhWht02.WAV
 268=Terran\PHOENIX\TPhWht03.WAV
 269=Terran\PHOENIX\TPhYes00.WAV
 270=Terran\PHOENIX\TPhYes01.WAV
 271=Terran\PHOENIX\TPhYes02.WAV
 272=Terran\PHOENIX\TPhYes03.WAV
 273=Terran\PHOENIX\TPhClo00.WAV
 274=Terran\PHOENIX\TPhClo01.WAV
 275=Terran\MARINE\TMaRdy00.WAV
 276=Terran\MARINE\TMaDth00.WAV
 277=Terran\MARINE\TMaDth01.WAV
 278=Terran\MARINE\TMaSti00.WAV
 279=Terran\MARINE\TMaSti01.WAV
 280=Terran\MARINE\TMaPss00.WAV
 281=Terran\MARINE\TMaPss01.WAV
 282=Terran\MARINE\TMaPss02.WAV
 283=Terran\MARINE\TMaPss03.WAV
 284=Terran\MARINE\TMaPss04.WAV
 285=Terran\MARINE\TMaPss05.WAV
 286=Terran\MARINE\TMaPss06.WAV
 287=Terran\MARINE\TMaWht00.WAV
 288=Terran\MARINE\TMaWht01.WAV
 289=Terran\MARINE\TMaWht02.WAV
 290=Terran\MARINE\TMaWht03.WAV
 291=Terran\MARINE\TMaYes00.WAV
 292=Terran\MARINE\TMaYes01.WAV
 293=Terran\MARINE\TMaYes02.WAV
 294=Terran\MARINE\TMaYes03.WAV
 295=Terran\Firebat\TFBRdy00.WAV
 296=Terran\Firebat\TFBDth00.WAV
 297=Terran\Firebat\TFBDth01.WAV
 298=Terran\Firebat\TFBDth02.WAV
 299=Terran\Firebat\TFBWht00.WAV
 300=Terran\Firebat\TFBWht01.WAV
 301=Terran\Firebat\TFBWht02.WAV
 302=Terran\Firebat\TFBWht03.WAV
 303=Terran\Firebat\TFBPss00.WAV
 304=Terran\Firebat\TFBPss01.WAV
 305=Terran\Firebat\TFBPss02.WAV
 306=Terran\Firebat\TFBPss03.WAV
 307=Terran\Firebat\TFBPss04.WAV
 308=Terran\Firebat\TFBPss05.WAV
 309=Terran\Firebat\TFBPss06.WAV
 310=Terran\Firebat\TFBYes00.WAV
 311=Terran\Firebat\TFBYes01.WAV
 312=Terran\Firebat\TFBYes02.WAV
 313=Terran\Firebat\TFBYes03.WAV
 314=Terran\Firebat\TFBFir00.WAV
 315=Terran\Firebat\TFBFir01.WAV
 316=Terran\TANK\TTaRdy00.WAV
 317=Terran\TANK\TTaDth00.WAV
 318=Terran\TANK\TTaTra00.WAV
 319=Terran\TANK\TTaTra01.WAV
 320=Terran\TANK\TTaPss00.WAV
 321=Terran\TANK\TTaPss01.WAV
 322=Terran\TANK\TTaPss02.WAV
 323=Terran\TANK\TTaPss03.WAV
 324=Terran\TANK\TTaWht00.WAV
 325=Terran\TANK\TTaWht01.WAV
 326=Terran\TANK\TTaWht02.WAV
 327=Terran\TANK\TTaWht03.WAV
 328=Terran\TANK\TTaYes00.WAV
 329=Terran\TANK\TTaYes01.WAV
 330=Terran\TANK\TTaYes02.WAV
 331=Terran\TANK\TTaYes03.WAV
 332=Terran\VESSEL\TVeRdy00.WAV
 333=Terran\VESSEL\TVePss00.WAV
 334=Terran\VESSEL\TVePss01.WAV
 335=Terran\VESSEL\TVePss02.WAV
 336=Terran\VESSEL\TVePss03.WAV
 337=Terran\VESSEL\TVePss04.WAV
 338=Terran\VESSEL\TVePss05.WAV
 339=Terran\VESSEL\TVePss06.WAV
 340=Terran\VESSEL\TVeWht00.WAV
 341=Terran\VESSEL\TVeWht01.WAV
 342=Terran\VESSEL\TVeWht02.WAV
 343=Terran\VESSEL\TVeWht03.WAV
 344=Terran\VESSEL\TVeYes00.WAV
 345=Terran\VESSEL\TVeYes01.WAV
 346=Terran\VESSEL\TVeYes02.WAV
 347=Terran\VESSEL\TVeYes03.WAV
 348=Terran\VESSEL\TVeDth00.WAV
 349=Terran\VESSEL\TVeDef00.WAV
 350=Terran\VESSEL\TVeEmp00.WAV
 351=Terran\VESSEL\TVeIrr00.WAV
 352=Terran\VULTURE\TVuRdy00.WAV
 353=Terran\VULTURE\TVuDth00.WAV
 354=Terran\VULTURE\TVuMin01.WAV
 355=Terran\VULTURE\TVuMin00.WAV
 356=Terran\VULTURE\TVuPss00.WAV
 357=Terran\VULTURE\TVuPss01.WAV
 358=Terran\VULTURE\TVuPss02.WAV
 359=Terran\VULTURE\TVuPss03.WAV
 360=Terran\VULTURE\TVuWht00.WAV
 361=Terran\VULTURE\TVuWht01.WAV
 362=Terran\VULTURE\TVuWht02.WAV
 363=Terran\VULTURE\TVuWht03.WAV
 364=Terran\VULTURE\TVuYes00.WAV
 365=Terran\VULTURE\TVuYes01.WAV
 366=Terran\VULTURE\TVuYes02.WAV
 367=Terran\VULTURE\TVuYes03.WAV
 368=Terran\SCV\TSCRdy00.WAV
 369=Terran\SCV\TSCDth00.WAV
 370=Terran\SCV\TSCPss00.WAV
 371=Terran\SCV\TSCPss01.WAV
 372=Terran\SCV\TSCPss02.WAV
 373=Terran\SCV\TSCPss03.WAV
 374=Terran\SCV\TSCPss04.WAV
 375=Terran\SCV\TSCPss05.WAV
 376=Terran\SCV\TSCPss06.WAV
 377=Terran\SCV\TSCWht00.WAV
 378=Terran\SCV\TSCWht01.WAV
 379=Terran\SCV\TSCWht02.WAV
 380=Terran\SCV\TSCWht03.WAV
 381=Terran\SCV\TSCYes00.WAV
 382=Terran\SCV\TSCYes01.WAV
 383=Terran\SCV\TSCYes02.WAV
 384=Terran\SCV\TSCYes03.WAV
 385=Terran\bldg\tacwht00.WAV
 386=Terran\bldg\tclwht00.WAV
 387=Terran\bldg\tcswh00.WAV
 388=Terran\bldg\tcsSca00.WAV
 389=Terran\bldg\tplwht00.WAV
 390=Terran\bldg\tglwht00.WAV
 391=Misc\UICWht00.WAV
 392=Terran\bldg\tmswht00.WAV
 393=Terran\bldg\tmtwht00.WAV
 394=Misc\UNRWht00.WAV
 395=Terran\bldg\tRLWht00.WAV
 396=Terran\bldg\tnswht00.WAV
 397=Terran\bldg\tpgwh00.WAV
 398=Terran\bldg\tplwht00.WAV
 399=Misc\button.WAV
 400=Terran\bldg\trewht00.WAV
 401=Terran\bldg\trfwht00.WAV
 402=Terran\bldg\tpwht00.WAV
 403=Terran\bldg\OnFirLrg.WAV
 404=Terran\SCV\TSCTra00.WAV
 405=Terran\RAYNORM\URaDth00.WAV
 406=Terran\RAYNORM\URaDth01.WAV
 407=Terran\RAYNORM\URaPss00.WAV
 408=Terran\RAYNORM\URaPss01.WAV
 409=Terran\RAYNORM\URaPss02.WAV
 410=Terran\RAYNORM\URaPss03.WAV
 411=Terran\RAYNORM\URaWht00.WAV
 412=Terran\RAYNORM\URaWht01.WAV
 413=Terran\RAYNORM\URaWht02.WAV
 414=Terran\RAYNORM\URaWht03.WAV
 415=Terran\RAYNORM\URaYes00.WAV
 416=Terran\RAYNORM\URaYes01.WAV
 417=Terran\RAYNORM\URaYes02.WAV
 418=Terran\RAYNORM\URaYes03.WAV
 419=Terran\RAYNORV\URVPss00.WAV
 420=Terran\RAYNORV\URVPss01.WAV
 421=Terran\RAYNORV\URVPss02.WAV
 422=Terran\RAYNORV\URVPss03.WAV
 423=Terran\RAYNORV\URVWht00.WAV
 424=Terran\RAYNORV\URVWht01.WAV
 425=Terran\RAYNORV\URVWht02.WAV
 426=Terran\RAYNORV\URVWht03.WAV
 427=Terran\RAYNORV\URVYes00.WAV
 428=Terran\RAYNORV\URVYes01.WAV
 429=Terran\RAYNORV\URVYes02.WAV
 430=Terran\RAYNORV\URVYes03.WAV
 431=Terran\DUKET\UDTPss00.WAV
 432=Terran\DUKET\UDTPss01.WAV
 433=Terran\DUKET\UDTPss02.WAV
 434=Terran\DUKET\UDTPss03.WAV
 435=Terran\DUKET\UDTPss04.WAV
 436=Terran\DUKET\UDTWht00.WAV
 437=Terran\DUKET\UDTWht01.WAV
 438=Terran\DUKET\UDTWht02.WAV
 439=Terran\DUKET\UDTWht03.WAV
 440=Terran\DUKET\UDTYes00.WAV
 441=Terran\DUKET\UDTYes01.WAV
 442=Terran\DUKET\UDTYes02.WAV
 443=Terran\DUKET\UDTYes03.WAV
 444=Terran\DUKEB\UDupss00.WAV
 445=Terran\DUKEB\UDupss01.WAV
 446=Terran\DUKEB\UDupss02.WAV
 447=Terran\DUKEB\UDupss03.WAV
 448=Terran\DUKEB\UDupss04.WAV
 449=Terran\DUKEB\UDuwht00.WAV
 450=Terran\DUKEB\UDuwht01.WAV
 451=Terran\DUKEB\UDuwht02.WAV
 452=Terran\DUKEB\UDuwht03.WAV
 453=Terran\DUKEB\UDuyes00.WAV
 454=Terran\DUKEB\UDuyes01.WAV
 455=Terran\DUKEB\UDuyes02.WAV
 456=Terran\DUKEB\UDuyes03.WAV
 457=Terran\KERRIGAN\UKePss00.WAV
 458=Terran\KERRIGAN\UKePss01.WAV
 459=Terran\KERRIGAN\UKePss02.WAV
 460=Terran\KERRIGAN\UKePss03.WAV
 461=Terran\KERRIGAN\UKePss04.WAV
 462=Terran\KERRIGAN\UKeWht00.WAV
 463=Terran\KERRIGAN\UKeWht01.WAV
 464=Terran\KERRIGAN\UKeWht02.WAV
 465=Terran\KERRIGAN\UKeWht03.WAV
 466=Terran\KERRIGAN\UKeYes00.WAV
 467=Terran\KERRIGAN\UKeYes01.WAV
 468=Terran\KERRIGAN\UKeYes02.WAV
 469=Terran\KERRIGAN\UKeYes03.WAV
 470=Terran\KERRIGAN\UKedth00.WAV
 471=Misc\Liftoff.WAV
 472=Misc\Land.WAV
 473=Protoss\Bldg\pAcWht00.WAV
 474=Protoss\Bldg\pAsWht00.WAV
 475=Protoss\Bldg\pBaWht00.WAV
 476=Protoss\Bldg\pBeWht00.WAV
 477=Protoss\Bldg\pCiWht00.WAV
 478=Protoss\Bldg\pFoWht00.WAV
 479=Protoss\Bldg\pGaWht00.WAV
 480=Protoss\Bldg\pGcWht00.WAV
 481=Protoss\Bldg\pBaAct00.WAV
 482=Misc\Button.WAV
 483=Protoss\Bldg\pNaWht00.WAV
 484=Protoss\Bldg\pNeWht00.WAV
 485=Protoss\Bldg\pPBWht00.WAV
 486=Misc\UtmWht00.WAV
 487=Protoss\Bldg\pPyWht00.WAV
 488=Protoss\Bldg\pRoWht00.WAV
 489=Misc\UPiWht00.WAV
 490=Protoss\Bldg\pTrWht00.WAV
 491=Protoss\Bldg\pWaWht00.WAV
 492=Protoss\DRAGOON\pDrRdy00.WAV
 493=Protoss\DRAGOON\pDrDth00.WAV
 494=Protoss\DRAGOON\pDrPss00.WAV
 495=Protoss\DRAGOON\pDrPss01.WAV

496=Protoss\DRAGOON\PD rPss02.WAV
 497=Protoss\DRAGOON\PD rPss03.WAV
 498=Protoss\DRAGOON\PD rWht00.WAV
 499=Protoss\DRAGOON\PD rWht01.WAV
 500=Protoss\DRAGOON\PD rWht02.WAV
 501=Protoss\DRAGOON\PD rWht03.WAV
 502=Protoss\DRAGOON\PD rWht04.WAV
 503=Protoss\DRAGOON\PD rWht05.WAV
 504=Protoss\DRAGOON\PD rWht06.WAV
 505=Protoss\DRAGOON\PD rWht07.WAV
 506=Protoss\DRAGOON\PD rYes00.WAV
 507=Protoss\DRAGOON\PD rYes01.WAV
 508=Protoss\DRAGOON\PD rYes02.WAV
 509=Protoss\DRAGOON\PD rYes03.WAV
 510=Protoss\DRAGOON\PD rYes04.WAV
 511=Protoss\DRAGOON\PD rYes05.WAV
 512=Protoss\DRAGOON\PD rYes06.WAV
 513=Protoss\SHUTTLE\PS hRdy00.WAV
 514=Protoss\SHUTTLE\PS hDth00.WAV
 515=Protoss\SHUTTLE\PS hPss00.WAV
 516=Protoss\SHUTTLE\PS hPss01.WAV
 517=Protoss\SHUTTLE\PS hPss02.WAV
 518=Protoss\SHUTTLE\PS hPss03.WAV
 519=Protoss\SHUTTLE\PS hPss04.WAV
 520=Protoss\SHUTTLE\PS hWht00.WAV
 521=Protoss\SHUTTLE\PS hWht01.WAV
 522=Protoss\SHUTTLE\PS hWht02.WAV
 523=Protoss\SHUTTLE\PS hWht03.WAV
 524=Protoss\SHUTTLE\PS hYes00.WAV
 525=Protoss\SHUTTLE\PS hYes01.WAV
 526=Protoss\SHUTTLE\PS hYes02.WAV
 527=Protoss\SHUTTLE\PS hYes03.WAV
 528=Protoss\SHUTTLE\PS hBld00.WAV
 529=Protoss\SHUTTLE\PS hBld03.WAV
 530=Protoss\SHUTTLE\PS hBld01.WAV
 531=Protoss\SHUTTLE\PS hBld02.WAV
 532=Protoss\SHUTTLE\PS hBld04.WAV
 533=Protoss\Scout\PS cDth00.WAV
 534=Protoss\Scout\PS cRdy00.WAV
 535=Protoss\Scout\PS cPss00.WAV
 536=Protoss\Scout\PS cPss01.WAV
 537=Protoss\Scout\PS cPss02.WAV
 538=Protoss\Scout\PS cPss03.WAV
 539=Protoss\Scout\PS cPss04.WAV
 540=Protoss\Scout\PS cWht00.WAV
 541=Protoss\Scout\PS cWht01.WAV
 542=Protoss\Scout\PS cWht02.WAV
 543=Protoss\Scout\PS cWht03.WAV
 544=Protoss\Scout\PS cYes00.WAV
 545=Protoss\Scout\PS cYes01.WAV
 546=Protoss\Scout\PS cYes02.WAV
 547=Protoss\Scout\PS cYes03.WAV
 548=Protoss\ARBITER\PA bCag00.WAV
 549=Protoss\ARBITER\PA bRdy00.WAV
 550=Protoss\ARBITER\PA bFo100.WAV
 551=Protoss\ARBITER\PA bFo101.WAV
 552=Protoss\ARBITER\PA bFo102.WAV
 553=Protoss\ARBITER\PA bFo103.WAV
 554=Protoss\ARBITER\PA bYes00.WAV
 555=Protoss\ARBITER\PA bPss01.WAV
 556=Protoss\ARBITER\PA bPss02.WAV
 557=Protoss\ARBITER\PA bPss03.WAV
 558=Protoss\ARBITER\PA bPss04.WAV
 559=Protoss\ARBITER\PA bWht00.WAV
 560=Protoss\ARBITER\PA bWht01.WAV
 561=Protoss\ARBITER\PA bWht02.WAV
 562=Protoss\ARBITER\PA bWht03.WAV
 563=Protoss\ARBITER\PA bYes01.WAV
 564=Protoss\ARBITER\PA bYes02.WAV
 565=Protoss\ARBITER\PA bYes03.WAV
 566=Protoss\ARBITER\PA bDth00.WAV
 567=Protoss\ARCHON\PA rRdy00.WAV
 568=Protoss\ARCHON\PA rMin00.WAV
 569=Protoss\ARCHON\PA rPss00.WAV
 570=Protoss\ARCHON\PA rPss01.WAV
 571=Protoss\ARCHON\PA rPss02.WAV
 572=Protoss\ARCHON\PA rPss03.WAV
 573=Protoss\ARCHON\PA rWht00.WAV
 574=Protoss\ARCHON\PA rWht01.WAV
 575=Protoss\ARCHON\PA rWht02.WAV
 576=Protoss\ARCHON\PA rWht03.WAV
 577=Protoss\ARCHON\PA rYes00.WAV
 578=Protoss\ARCHON\PA rYes01.WAV
 579=Protoss\ARCHON\PA rYes02.WAV
 580=Protoss\ARCHON\PA rYes03.WAV
 581=Protoss\ARCHON\PA rDth00.WAV
 582=Protoss\CARRIER\PC aRdy00.WAV
 583=Protoss\CARRIER\PC aPss00.WAV
 584=Protoss\CARRIER\PC aPss01.WAV
 585=Protoss\CARRIER\PC aPss02.WAV
 586=Protoss\CARRIER\PC aPss03.WAV
 587=Protoss\CARRIER\PC aWht00.WAV
 588=Protoss\CARRIER\PC aWht01.WAV
 589=Protoss\CARRIER\PC aWht02.WAV
 590=Protoss\CARRIER\PC aWht03.WAV
 591=Protoss\CARRIER\PC aYes00.WAV
 592=Protoss\CARRIER\PC aYes01.WAV
 593=Protoss\CARRIER\PC aYes02.WAV
 594=Protoss\CARRIER\PC aYes03.WAV
 595=Protoss\CARRIER\PC aDth00.WAV
 596=Protoss\CARRIER\PC aDth01.WAV
 597=Protoss\PROBE\PP rRdy00.WAV
 598=Protoss\PROBE\PP rDth00.WAV
 599=Protoss\PROBE\PP rAtt00.WAV
 600=Protoss\PROBE\PP rAtt01.WAV
 601=Protoss\PROBE\PP rMin00.WAV
 602=Protoss\PROBE\PP rPss00.WAV
 603=Protoss\PROBE\PP rPss01.WAV
 604=Protoss\PROBE\PP rPss02.WAV
 605=Protoss\PROBE\PP rPss03.WAV
 606=Protoss\PROBE\PP rWht00.WAV
 607=Protoss\PROBE\PP rWht01.WAV
 608=Protoss\PROBE\PP rWht02.WAV
 609=Protoss\PROBE\PP rWht03.WAV
 610=Protoss\PROBE\PP rYes00.WAV
 611=Protoss\PROBE\PP rYes01.WAV
 612=Protoss\PROBE\PP rYes02.WAV
 613=Protoss\PROBE\PP rYes03.WAV
 614=Protoss\INTERCEPT\PI nLau00.WAV
 615=Protoss\PROBE\PP rPss00.WAV
 616=Protoss\TEMPLAR\PT eSum00.WAV
 617=Protoss\TEMPLAR\PT eSum00.WAV
 618=Protoss\TEMPLAR\PT eHal00.WAV
 619=Protoss\TEMPLAR\PT eHal01.WAV
 620=Protoss\TEMPLAR\PT eSto00.WAV
 621=Protoss\TEMPLAR\PT eSto01.WAV
 622=Protoss\TEMPLAR\PT eRdy00.WAV
 623=Protoss\TEMPLAR\PT ePss00.WAV
 624=Protoss\TEMPLAR\PT ePss01.WAV
 625=Protoss\TEMPLAR\PT ePss02.WAV
 626=Protoss\TEMPLAR\PT ePss03.WAV
 627=Protoss\TEMPLAR\PT eWht00.WAV
 628=Protoss\TEMPLAR\PT eWht01.WAV
 629=Protoss\TEMPLAR\PT eWht02.WAV
 630=Protoss\TEMPLAR\PT eWht03.WAV
 631=Protoss\TEMPLAR\PT eYes00.WAV
 632=Protoss\TEMPLAR\PT eYes01.WAV
 633=Protoss\TEMPLAR\PT eYes02.WAV
 634=Protoss\TEMPLAR\PT eYes03.WAV
 635=Protoss\TEMPLAR\PT eDth00.WAV
 636=Protoss\TEMPLAR\PT eMov00.WAV
 637=Protoss\TRILOBYTE\PT rRdy00.WAV
 638=Protoss\TRILOBYTE\PT rDth00.WAV
 639=Protoss\TRILOBYTE\PT rPss00.WAV
 640=Protoss\TRILOBYTE\PT rPss01.WAV
 641=Protoss\TRILOBYTE\PT rPss02.WAV
 642=Protoss\TRILOBYTE\PT rWht00.WAV
 643=Protoss\TRILOBYTE\PT rWht01.WAV
 644=Protoss\TRILOBYTE\PT rWht02.WAV
 645=Protoss\TRILOBYTE\PT rWht03.WAV
 646=Protoss\TRILOBYTE\PT rYes00.WAV
 647=Protoss\TRILOBYTE\PT rYes01.WAV
 648=Protoss\TRILOBYTE\PT rYes02.WAV
 649=Protoss\TRILOBYTE\PT rYes03.WAV
 650=Protoss\Witness\PW iRdy00.WAV
 651=Protoss\Witness\PW iDth00.WAV
 652=Protoss\Witness\PW iDth01.WAV
 653=Protoss\Witness\PW iPss00.WAV
 654=Protoss\Witness\PW iPss01.WAV
 655=Protoss\Witness\PW iPss02.WAV
 656=Protoss\Witness\PW iPss03.WAV
 657=Protoss\Witness\PW iPss04.WAV
 658=Protoss\Witness\PW iWht00.WAV
 659=Protoss\Witness\PW iWht01.WAV
 660=Protoss\Witness\PW iYes00.WAV
 661=Protoss\Witness\PW iYes01.WAV
 662=Protoss\ZEALOT\pzeAtt00.WAV
 663=Protoss\ZEALOT\pzeAtt01.WAV
 664=Protoss\ZEALOT\pzeHit00.WAV
 665=Protoss\ZEALOT\pzeRag00.WAV
 666=Protoss\ZEALOT\pzeRdy00.WAV
 667=Protoss\ZEALOT\pzePss00.WAV
 668=Protoss\ZEALOT\pzePss01.WAV
 669=Protoss\ZEALOT\pzePss02.WAV
 670=Protoss\ZEALOT\pzeWht00.WAV
 671=Protoss\ZEALOT\pzeWht01.WAV
 672=Protoss\ZEALOT\pzeWht02.WAV
 673=Protoss\ZEALOT\pzeWht03.WAV
 674=Protoss\ZEALOT\pzeYes00.WAV
 675=Protoss\ZEALOT\pzeYes01.WAV
 676=Protoss\ZEALOT\pzeYes02.WAV
 677=Protoss\ZEALOT\pzeYes03.WAV
 678=Protoss\ZEALOT\pzeDth00.WAV
 679=Protoss\FENIXD\UFD Pss00.WAV
 680=Protoss\FENIXD\UFD Pss01.WAV
 681=Protoss\FENIXD\UFD Pss02.WAV
 682=Protoss\FENIXD\UFD Pss03.WAV
 683=Protoss\FENIXD\UFD Wht00.WAV
 684=Protoss\FENIXD\UFD Wht01.WAV
 685=Protoss\FENIXD\UFD Wht02.WAV
 686=Protoss\FENIXD\UFD Wht03.WAV
 687=Protoss\FENIXD\UFD Yes00.WAV
 688=Protoss\FENIXD\UFD Yes01.WAV
 689=Protoss\FENIXD\UFD Yes02.WAV
 690=Protoss\FENIXD\UFD Yes03.WAV
 691=Protoss\FENIXZ\UFE Pss00.WAV
 692=Protoss\FENIXZ\UFE Pss01.WAV
 693=Protoss\FENIXZ\UFE Pss02.WAV
 694=Protoss\FENIXZ\UFE Pss03.WAV
 695=Protoss\FENIXZ\UFE Wht00.WAV
 696=Protoss\FENIXZ\UFE Wht01.WAV
 697=Protoss\FENIXZ\UFE Wht02.WAV
 698=Protoss\FENIXZ\UFE Wht03.WAV
 699=Protoss\FENIXZ\UFE Yes00.WAV
 700=Protoss\FENIXZ\UFE Yes01.WAV
 701=Protoss\FENIXZ\UFE Yes02.WAV
 702=Protoss\FENIXZ\UFE Yes03.WAV
 703=Protoss\GANTRITHOR\UTC Pss00.WAV
 704=Protoss\GANTRITHOR\UTC Pss01.WAV
 705=Protoss\GANTRITHOR\UTC Pss02.WAV
 706=Protoss\GANTRITHOR\UTC Pss03.WAV
 707=Protoss\GANTRITHOR\UTC Wht00.WAV
 708=Protoss\GANTRITHOR\UTC Wht01.WAV
 709=Protoss\GANTRITHOR\UTC Wht02.WAV
 710=Protoss\GANTRITHOR\UTC Wht03.WAV
 711=Protoss\GANTRITHOR\UTC Yes00.WAV
 712=Protoss\GANTRITHOR\UTC Yes01.WAV
 713=Protoss\GANTRITHOR\UTC Yes02.WAV
 714=Protoss\GANTRITHOR\UTC Yes03.WAV
 715=Protoss\TASSADAR\UTa Pss00.WAV
 716=Protoss\TASSADAR\UTa Pss01.WAV
 717=Protoss\TASSADAR\UTa Pss02.WAV

860=Zerg\Guardian\ZGuWht02.WAV
861=Zerg\Guardian\ZGuWht03.WAV
862=Zerg\Guardian\ZGuYes00.WAV
863=Zerg\Guardian\ZGuYes01.WAV
864=Zerg\Guardian\ZGuYes02.WAV
865=Zerg\Guardian\ZGuYes03.WAV
866=Zerg\Hydra\ZHyRdy00.WAV
867=Zerg\Hydra\ZHyDth00.WAV
868=Zerg\Hydra\ZHyPss00.WAV
869=Zerg\Hydra\ZHyPss01.WAV
870=Zerg\Hydra\ZHyWht00.WAV
871=Zerg\Hydra\ZHyWht01.WAV
872=Zerg\Hydra\ZHyWht02.WAV
873=Zerg\Hydra\ZHyYes00.WAV
874=Zerg\Hydra\ZHyYes01.WAV
875=Zerg\Hydra\ZHyYes02.WAV
876=Zerg\Hydra\ZHyYes03.WAV
877=Zerg\Ultra\ZUIRor00.WAV
878=Zerg\Ultra\ZUIDth00.WAV
879=Zerg\Ultra\ZUIPss00.WAV
880=Zerg\Ultra\ZUIPss01.WAV
881=Zerg\Ultra\ZUIPss02.WAV
882=Zerg\Ultra\ZUIRdy00.WAV
883=Zerg\Ultra\ZUIWht00.WAV
884=Zerg\Ultra\ZUIWht01.WAV
885=Zerg\Ultra\ZUIWht02.WAV
886=Zerg\Ultra\ZUIWht03.WAV
887=Zerg\Ultra\ZUIYes00.WAV
888=Zerg\Ultra\ZUIYes01.WAV
889=Zerg\Ultra\ZUIYes02.WAV
890=Zerg\Ultra\ZUIYes03.WAV
891=Zerg\Ultra\zulAtt00.WAV
892=Zerg\Ultra\zulAtt01.WAV
893=Zerg\Ultra\zulAtt02.WAV
894=Zerg\Ultra\zulHit00.WAV
895=Zerg\Ultra\zulHit01.WAV
896=Zerg\Zergling\ZZeDth00.WAV
897=Zerg\Zergling\ZZePss00.WAV
898=Zerg\Zergling\ZZePss01.WAV
899=Zerg\Zergling\ZZePss02.WAV
900=Zerg\Zergling\ZZeRdy00.WAV
901=Zerg\Zergling\ZZeWht00.WAV
902=Zerg\Zergling\ZZeWht01.WAV
903=Zerg\Zergling\ZZeWht02.WAV
904=Zerg\Zergling\ZZeWht03.WAV
905=Zerg\Zergling\ZZeYes00.WAV
906=Zerg\Zergling\ZZeYes01.WAV
907=Zerg\Zergling\ZZeYes02.WAV
908=Zerg\Zergling\ZZeYes03.WAV
909=Zerg\OVERLORD\ZOvRdy00.WAV
910=Zerg\OVERLORD\ZOvDth00.WAV
911=Zerg\OVERLORD\ZOvPss00.WAV
912=Zerg\OVERLORD\ZOvWht00.WAV
913=Zerg\OVERLORD\ZOvWht01.WAV
914=Zerg\OVERLORD\ZOvWht02.WAV
915=Zerg\OVERLORD\ZOvWht03.WAV
916=Zerg\OVERLORD\ZOvYes00.WAV
917=Zerg\OVERLORD\ZOvYes01.WAV
918=Zerg\OVERLORD\ZOvYes02.WAV
919=Zerg\OVERLORD\ZOvYes03.WAV
920=Zerg\Queen\ZQuTag00.WAV
921=Zerg\Queen\ZQuTag01.WAV
922=Zerg\Queen\ZQuSwm00.WAV
923=Zerg\Queen\ZQuSwm01.WAV
924=Zerg\Queen\ZQuEns00.WAV
925=Zerg\Queen\ZQuDth00.WAV
926=Zerg\Queen\ZQuDth01.WAV
927=Zerg\Queen\ZQuDth02.WAV
928=Zerg\Queen\ZQuRdy00.WAV
929=Zerg\Queen\ZQuPss00.WAV
930=Zerg\Queen\ZQuPss01.WAV
931=Zerg\Queen\ZQuPss02.WAV
932=Zerg\Queen\ZQuPss03.WAV
933=Zerg\Queen\ZQuWht00.WAV
934=Zerg\Queen\ZQuWht01.WAV
935=Zerg\Queen\ZQuWht02.WAV
936=Zerg\Queen\ZQuWht03.WAV
937=Zerg\Queen\ZQuYes00.WAV

938=Zerg\Queen\ZQuYes01.WAV
 939=Zerg\Queen\ZQuYes02.WAV
 940=Zerg\Queen\ZQuYes03.WAV
 941=Zerg\Mutalid\ZMuRdy00.WAV
 942=Zerg\Mutalid\ZMuDth00.WAV
 943=Zerg\Mutalid\ZMuPss00.WAV
 944=Zerg\Mutalid\ZMuPss01.WAV
 945=Zerg\Mutalid\ZMuPss02.WAV
 946=Zerg\Mutalid\ZMuPss03.WAV
 947=Zerg\Mutalid\ZMuWht00.WAV
 948=Zerg\Mutalid\ZMuWht01.WAV
 949=Zerg\Mutalid\ZMuWht02.WAV
 950=Zerg\Mutalid\ZMuWht03.WAV
 951=Zerg\Mutalid\ZMuYes00.WAV
 952=Zerg\Mutalid\ZMuYes01.WAV
 953=Zerg\Mutalid\ZMuYes02.WAV
 954=Zerg\Mutalid\ZMuYes03.WAV
 955=Zerg\ZERGKERRI\UKiPss00.wav
 956=Zerg\ZERGKERRI\UKiPss01.wav
 957=Zerg\ZERGKERRI\UKiPss02.wav
 958=Zerg\ZERGKERRI\UKiPss03.wav
 959=Zerg\ZERGKERRI\UKiWht00.wav
 960=Zerg\ZERGKERRI\UKiWht01.wav
 961=Zerg\ZERGKERRI\UKiWht02.wav
 962=Zerg\ZERGKERRI\UKiWht03.wav
 963=Zerg\ZERGKERRI\UKiYes00.wav
 964=Zerg\ZERGKERRI\UKiYes01.wav
 965=Zerg\ZERGKERRI\UKiYes02.wav
 966=Zerg\ZERGKERRI\UKiYes03.wav
 967=Zerg\ZERGKERRI\UKiDth00.wav
 968=Misc\CRITTERS\PbWht01.wav
 969=Misc\CRITTERS\PbWht02.wav
 970=Misc\CRITTERS\PbWht03.wav
 971=Misc\CRITTERS\PbDeath01.wav
 972=Misc\CRITTERS\ScWht01.wav
 973=Misc\CRITTERS\ScWht02.wav
 974=Misc\CRITTERS\ScWht03.wav
 975=Misc\CRITTERS\ScDeath01.wav
 976=Misc\CRITTERS\TerWht01.wav
 977=Misc\CRITTERS\TerWht02.wav
 978=Misc\CRITTERS\TerWht03.wav
 979=Misc\CRITTERS\TerDeath01.wav
 980=Terran\Duran\TDnPss00.wav
 981=Terran\Duran\TDnPss01.wav
 982=Terran\Duran\TDnPss02.wav
 983=Terran\Duran\TDnPss03.wav
 984=Terran\Duran\TDnPss04.wav
 985=Terran\Duran\TDnPss05.wav
 986=Terran\Duran\TDnPss06.wav
 987=Terran\Duran\TDnPss07.wav
 988=Terran\Duran\TDnPss08.wav
 989=Terran\Duran\TDnWht00.wav
 990=Terran\Duran\TDnWht01.wav
 991=Terran\Duran\TDnWht02.wav
 992=Terran\Duran\TDnWht03.wav
 993=Terran\Duran\TDnYes00.wav
 994=Terran\Duran\TDnYes01.wav
 995=Terran\Duran\TDnYes02.wav
 996=Terran\Duran\TDnYes03.wav
 997=Terran\Duran\TDnDth00.wav
 998=Terran\Medic\TMdCure.wav
 999=Terran\Medic\TMdRdy00.wav
 1000=Terran\Medic\TMdDth00.wav
 1001=Terran\Medic\TMdWht00.wav
 1002=Terran\Medic\TMdWht01.WAV
 1003=Terran\Medic\TMdWht02.WAV
 1004=Terran\Medic\TMdWht03.wav
 1005=Terran\Medic\TMdYes00.wav
 1006=Terran\Medic\TMdYes01.wav
 1007=Terran\Medic\TMdYes02.wav
 1008=Terran\Medic\TMdYes03.wav
 1009=Terran\Medic\TMdPss00.wav
 1010=Terran\Medic\TMdPss01.WAV
 1011=Terran\Medic\TMdPss02.WAV
 1012=Terran\Medic\TMdPss03.WAV
 1013=Terran\Medic\TMdPss04.WAV
 1014=Terran\Medic\TMdPss05.WAV
 1015=Terran\Medic\TMdPss06.wav
 1016=Terran\Medic\TMdflsh.wav
 1017=Terran\Medic\TMdRest.wav
 1018=Terran\Medic\TMdHeal.wav
 1019=Terran\Medic\Miopia1.wav
 1020=Terran\Frigate\AfterOn.wav
 1021=Terran\Frigate\AfterOff.wav
 1022=Bullet\Tfrhit.wav
 1023=Bullet\Tfrshoot.wav
 1024=Terran\FRIGATE\TVkRdy00.WAV
 1025=Terran\FRIGATE\TVkPss00.WAV
 1026=Terran\FRIGATE\TVkPss01.WAV
 1027=Terran\FRIGATE\TVkPss02.WAV
 1028=Terran\FRIGATE\TVkPss03.WAV
 1029=Terran\FRIGATE\TVkPss04.WAV
 1030=Terran\FRIGATE\TVkPss05.WAV
 1031=Terran\FRIGATE\TVkWh00.WAV
 1032=Terran\FRIGATE\TVkWh01.WAV
 1033=Terran\FRIGATE\TVkWh02.WAV
 1034=Terran\FRIGATE\TVkWh03.WAV
 1035=Terran\FRIGATE\TVkYes00.WAV
 1036=Terran\FRIGATE\TVkYes01.WAV
 1037=Terran\FRIGATE\TVkYes02.WAV
 1038=Terran\FRIGATE\TVkYes03.WAV
 1039=Terran\FRIGATE\TVkYes04.WAV
 1040=Terran\FRIGATE\TVkDth00.WAV
 1041=Protoss\Corsair\PCoRdy00.wav
 1042=Protoss\Corsair\PCorhit1.wav
 1043=Protoss\Corsair\PCorhit2.wav
 1044=Protoss\Corsair\PCoWht00.wav
 1045=Protoss\Corsair\PCoWht01.WAV
 1046=Protoss\Corsair\PCoWht02.WAV
 1047=Protoss\Corsair\PCoWht03.wav
 1048=Protoss\Corsair\PCoYes00.wav
 1049=Protoss\Corsair\PCoYes01.WAV
 1050=Protoss\Corsair\PCoYes02.WAV
 1051=Protoss\Corsair\PCoYes03.wav
 1052=Protoss\Corsair\PCoPss00.wav
 1053=Protoss\Corsair\PCoPss01.WAV
 1054=Protoss\Corsair\PCoPss02.WAV
 1055=Protoss\Corsair\PCoPss03.WAV
 1056=Protoss\Corsair\PCoPss04.WAV
 1057=Protoss\Corsair\PCoPss05.WAV
 1058=Protoss\Corsair\PCoPss06.wav
 1059=Protoss\Corsair\PCorWeb1.wav
 1060=Protoss\Corsair\PCorlasr2.wav
 1061=Protoss\Darchon\feedback.wav
 1062=Protoss\Darchon\mind.wav
 1063=Protoss\Darchon\ParaAttk.wav
 1064=Protoss\Darchon\Parahit.wav
 1065=Protoss\DARCHON\PDaRdy00.WAV
 1066=Protoss\DARCHON\PDaDth00.WAV
 1067=Protoss\DARCHON\PDaPss00.WAV
 1068=Protoss\DARCHON\PDaPss01.WAV
 1069=Protoss\DARCHON\PDaPss02.WAV
 1070=Protoss\DARCHON\PDaPss03.WAV
 1071=Protoss\DARCHON\PDaWht00.WAV
 1072=Protoss\DARCHON\PDaWht01.WAV
 1073=Protoss\DARCHON\PDaWht02.WAV
 1074=Protoss\DARCHON\PDaWht03.WAV
 1075=Protoss\DARCHON\PDaYes00.WAV
 1076=Protoss\DARCHON\PDaYes01.WAV
 1077=Protoss\DARCHON\PDaYes02.WAV
 1078=Protoss\DARCHON\PDaYes03.WAV
 1079=Zerg\LURKER\ZLuRdy00.WAV
 1080=Zerg\LURKER\ZLuDth00.WAV
 1081=Zerg\LURKER\ZLuPss00.WAV
 1082=Zerg\Lurker\ZLuPss01.WAV
 1083=Zerg\Lurker\ZLuPss02.WAV
 1084=Zerg\LURKER\ZLuPss03.WAV
 1085=Zerg\LURKER\ZLuWht00.WAV
 1086=Zerg\LURKER\ZLuWht01.WAV
 1087=Zerg\LURKER\ZLuWht02.WAV
 1088=Zerg\LURKER\ZLuWht03.WAV
 1089=Zerg\LURKER\ZLuYes00.WAV
 1090=Zerg\LURKER\ZLuYes01.WAV
 1091=Zerg\LURKER\ZLuYes02.WAV
 1092=Zerg\LURKER\ZLuYes03.WAV
 1093=Zerg\LURKER\ZLuBurw.wav
 1094=Zerg\Devourer\firesuck.wav
 1095=Zerg\Devourer\goophit.wav
 1096=Zerg\Devourer\ZDvRdy00.WAV
 1097=Zerg\Devourer\ZDvDth00.WAV
 1098=Zerg\Devourer\ZDvPss00.WAV
 1099=Zerg\Devourer\ZDvPss01.WAV
 1100=Zerg\Devourer\ZDvPss02.WAV
 1101=Zerg\Devourer\ZDvWht00.WAV
 1102=Zerg\Devourer\ZDvWht01.WAV
 1103=Zerg\Devourer\ZDvWht02.WAV
 1104=Zerg\Devourer\ZDvYes00.WAV
 1105=Zerg\Devourer\ZDvYes01.WAV
 1106=Zerg\Devourer\ZDvYes02.WAV
 1107=Zerg\Devourer\ZDvYes03.WAV
 1108=Bullet\ZLrkFir1.wav
 1109=Bullet\ZLrkFir2.wav
 1110=Bullet\ZLrkHit1.wav
 1111=Bullet\ZLrkHit2.wav
 1112=Zerg\ZergDuran\ZDnPss00.wav
 1113=Zerg\ZergDuran\ZDnPss01.wav
 1114=Zerg\ZergDuran\ZDnPss02.wav
 1115=Zerg\ZergDuran\ZDnPss03.wav
 1116=Zerg\ZergDuran\ZDnPss04.wav
 1117=Zerg\ZergDuran\ZDnPss05.wav
 1118=Zerg\ZergDuran\ZDnPss06.wav
 1119=Zerg\ZergDuran\ZDnPss07.wav
 1120=Zerg\ZergDuran\ZDnPss08.wav
 1121=Zerg\ZergDuran\ZDnWht00.wav
 1122=Zerg\ZergDuran\ZDnWht01.wav
 1123=Zerg\ZergDuran\ZDnWht02.wav
 1124=Zerg\ZergDuran\ZDnWht03.wav
 1125=Zerg\ZergDuran\ZDnYes00.wav
 1126=Zerg\ZergDuran\ZDnYes01.wav
 1127=Zerg\ZergDuran\ZDnYes02.wav
 1128=Zerg\ZergDuran\ZDnYes03.wav
 1129=Unknown 0
 1130=Unknown 1
 1131=Unknown 2
 1132=Unknown 3
 1133=Unknown 4
 1134=Unknown 5
 1135=Unknown 6
 1136=Unknown 7
 1137=Unknown 8
 1138=Unknown 9
 1139=Unknown 10
 1140=Unknown 11
 1141=Unknown 12
 1142=Unknown 13
 1143=Unknown 14
 1144=Unknown 15

Weapons.DAT:

000=Gauss Rifle	005=Fragmentation Grenade	010=Hellfire Missile Pack
001=Gauss Rifle	006=Spider Mines	011=Arclite Cannon
002=C-10 Canister Rifle	007=Twin Autocannons	012=Arclite Cannon
003=C-10 Canister Rifle	008=Hellfire Missile Pack	013=Fusion Cutter
004=Fragmentation Grenade	009=Twin Autocannons	014=Fusion Cutter

015=Gemini Missiles	054=Suicide	093=Undefined Weapon Name
016=Burst Lasers	055=Suicide	094=Undefined Weapon Name
017=Gemini Missiles	056=rParasite	095=Flechette Grenade
018=Burst Lasers	057=bSpawn Broodlings	096=Twin Autocannons
019=ATS Laser Battery	058=eEnsnare	097=Hellfire Missile Pack
020=ATA Laser Battery	059=wDark Swarm	098=Flame Thrower
021=ATS Laser Battery	060=gPlague	099=Hellfire Missile Pack
022=ATA Laser Battery	061=cConsume	100=Neutron Flare
023=ATS Laser Battery	062=Particle Beam	101=dDisruption Web
024=ATA Laser Battery	063=Particle Beam	102=rRestoration
025=Flame Thrower	064=Psi Blades	103=Halo Rockets
026=Flame Thrower	065=Psi Blades	104=Corrosive Acid
027=Arclite Shock Cannon	066=Phase Disruptor	105=Mind Control
028=Arclite Shock Cannon	067=Phase Disruptor	106=fFeedback
029=Longbolt Missile	068=Psi Assault	107=fOptical Flare
030=Yamato Gun	069=Psi Assault	108=Maelstrom
031=Nuclear Strike	070=Psionic Shockwave	109=Subterranean Spines
032=Lockdown	071=Psionic Shockwave	110=Gauss Rifle
033=EMP Shockwave	072=Unused	111=Warp Blades
034=iIrradiate	073=Dual Photon Blasters	112=C-10 Canister Rifle
035=Claws	074=Anti-matter Missiles	113=C-10 Canister Rifle
036=Claws	075=Dual Photon Blasters	114=Dual Photon Blasters
037=Claws	076=Anti-matter Missiles	115=Anti-matter Missiles
038=Needle Spines	077=Phase Disruptor Cannon	116=C-10 Canister Rifle
039=Needle Spines	078=Phase Disruptor Cannon	117=Gauss Rifle
040=Kaiser Blades	079=Pulse Cannon	118=Gauss Rifle
041=Kaiser Blades	080=STS Photon Cannon	119=Gauss Rifle
042=Toxic Spores	081=STA Photon Cannon	120=Gauss Rifle
043=Spines	082=Scarab	121=Gauss Rifle
044=Spines	083=tStasis Field	122=Gauss Rifle
045=Acid Spray	084=tPsionic Storm	123=Gauss Rifle
046=Acid Spore	085=Warp Blades	124=Gauss Rifle
047=Acid Spore	086=Warp Blades	125=Gauss Rifle
048=Glave Wurm	087=Missiles	126=Gauss Rifle
049=Glave Wurm	088=Laser Battery	127=Gauss Rifle
050=Venom	089=Tormentor Missiles	128=Gauss Rifle
051=Venom	090=Bombs	129=Gauss Rifle
052=Seeker Spores	091=Raider Gun	
053=Subterranean Tentacle	092=Undefined Weapon Name	