WarDraft 1.1

by Alexander Cech

Disclaimer: WarDraft is freeware, so you may do whatever you want with it.
The author assumes no liability for damages, direct or consequential, which may result from the use of WarDraft.

Introduction

Tools

Appendices

About WarDraft and WAR archives

WarDraft is an editor for the data archives used in Warcraft II. These are found in the subdirectory "DATA" of Warcraft's installation directory (usually: C\WAR2\DATA\) or in the CD's "DATA" directory and have an ".WAR" extension (exception: SFXDAT.SUD, the sound archive, and MUDDAT.CUD). They include the builtin levels, pictures, sound, text and much more.

This File	*includes*
MAINDAT:WAR	levels,some pictures, fonts, unit/building-,cursor-,map-graphics and some (yet) unknown stuff
REZDAT:WAR	mostly pictures
STRDAT:WAR	all kinds of text appearing in the game
SFXDAT:SUD	sound effects
SNDDAT:WAR	spoken mission briefings
MUDDAT:CUD	animated cutscenes

If you are interested in the WAR-format click here.

WarDraft is NOT a Level- or Map-Editor !
For constructing your own levels use the Editor included with Warcraft II.
I also recommend Daniel Lembergs War2xEd - a MUST for all PUD-creators
(http://www.bayserve.net/~lemberg/)

Knowledge bases

WarDraft does its best to identify each entry in the archives, but it can only give a general description (e.g. it can identify a PUD (a level), but it cannot tell what number it is), and it's sometimes (very rarely) guesses wrong, especially image entries (this is not a bug). That's where Knowledge Bases come in. These are simple text files, similarily to Windows .INI files, containing all the information I could find out about the archives. Beside a short description they also contain the palette-numbers for stored images (Image files are stored without palette information, so you usually have to either view them in grayscale or search for the corresponding palette entry).
Knowledge Bases basically identify the current archive by its filename. If you often work with a renamed file (say MAIN.WAR instead of MAINDAT.WAR) you can now choose the respective entry in the Knowledge Base manually. See *File/Select knowledge base entry*.

Warcraft II V1.2 users (those of you who don't have the Expansion CD):
The included Knowledge Base "WAR131.KNW" describes the archives of Version 1.31 (i.e. the expansion-updated ones).
The first time you use WarDraft you have to load "WAR12.KNW" (*File/(Re)load knowledge base*)

How to use WarDraft

First, use the *File/Open archive* command to open an existing archive. The main part of the editor window is filled with information about the diferent entries in the archive. To do something with an entry, you have to select it in this list first. Now most of the entries in the *Edit* menu (and the corresponding toolbar buttons) become active.

Information about the entries

For each entry in the archive there is a lot of information displayed. If some of it is not be interesting to you, simply resize the corresponding column until it can no longer be seen. By clicking on the column headers you can switch between hex and decimal display.
Number: The file number of the entry (it's position in the archive)
Offset: Where in the archive the entry-header starts.
Length: The space in the archive used by the entry (including it's header)
Orig.Length: The real size of the entry (uncompressed, without header)
Flags: Either CMP for "compressed" or blank
Description: What WarDraft thinks this entry is (or info from the Knowledge Base)

Command overview

General:
By double-clicking an entry you invoke the default command: images will be shown, text-tables listed, audio files and animations played, PUDs viewed. Everything else is shown as hexdump.

File menu
Edit menu
Options menu
Tools menu

File menu

Open
Reads information from an existing archive and displays it in the main window.

(Re)load knowledge base
Loads a selectable Knowledge Base. WarDraft remembers the filename of the Knowledge Base you load and uses it also when autoloading.

Select knowledge base entry
This is useful if you work with a copy of a war archive and there is no matching entry in the Knowledge Base. This submenu list all entries currently available. Just select one to use it.
Example: you are working with a copy of maindat.war, called maincopy.war. Now, since there is no entry in the Knowledge Base for a file called maincopy.war, you won't get any additional information. Now select maindat.war from the menu *File/Select knowledge base entry* and WarDraft will display all info it knows about maindat.war.

Exit
guess...

Edit menu

Extract file
Saves the selected entry as a stand-alone file (that's how you can extract PUDS and edit them with the level editor !)
If the entry is something that could be saved in different formats (like pictures, palettes, midi files) you can choose the format.
If the entry is compressed you also have the option to save it compressed or uncompressed.

Import file
This allows you to insert self-created or modified files into the archive (PUDs,images,sound,text....). You may optionally compress your file before inserting it into the archive - however I don't know if Warcraft II allows compression of any file. So it's best to use compression only if the entry you replace was also compressed.
A word of warning: always keep the entry-numbering as it was in the original archive; if you want to replace the title-screen for example (entry #299 in MAINDAT.WAR) make sure you REPLACE entry #299 with your image data. Although WarDraft allows you to ADD additional entries to the archive, these won't work with Warcraft II anymore. Only expert users should use the *insert before* or *insert after* commands. The same applies to the *Edit/Delete file* command.

Delete file
Deletes the selected file from the archive. Only for users who know what they do ! (see warning above)

Show as image data
Tries to view the selected entry as image data. The images are stored without palette information, so unless you have loaded a Knowledge Base (and it knows the corresponding palette-entry), you are given the option to either select a palette-entry manually or use a grayscale palette.
[See Appendix C - Known Bugs]

Show as hex dump
This displays a hex dump of the selected file. If it is compressed you may view it compressed or uncompressed.
[See Appendix C - Known Bugs]

Play WAV file
Plays the selected entry as a RIFF WAVE audio file. (Have a look at SFXDAT.SUD !)

Play MIDI file
Plays the selected entry as a MIDI sound file. (Many MIDIs are in MAINDAT.WAR !).
You can **instantly resume your work** while the sound plays in the background. Use the Play/Pause/Stop buttons at the upper right to control the playing.

Display/Edit texts
Displays and edits Text-entries, as found in STRDAT.WAR. For detailed instructions click here.

Play Smacker animation
Plays Smacker animation-entries. For this to work you need a smacker-compatible player (like SmackPlay from RAD-Software), and you need to tell WarDraft where to find it with *Options/Utility paths*.

Show PUD map
Displays the selected PUD (level) entry. For this to work you have to specify a path to a working copy of maindat.war in *Options/Utility Paths*.

Show graphics
Displays Units, Icons, Buildings, Missiles and similar stuff.

Options menu

Decompress automatically
If checked WarDraft doesn't ask you if you want to save or view entries in their compressed or decompressed form - if they are compressed they will automatically be decompressed before saving or viewing.

Autoload knowledge base
If checked, every time you open an archive the corresponding Knowledge Base is loaded too.

Ask for palette always
WarDraft usually doesnt asks for palette information only when viewing images which have an entry in the Knowledge Base and the corresponding palette is known too. By checking this option you will always be asked for palette information.

Utility paths
If you use external utilities with WarDraft (like the SmackPlay animation player by RAD Software), WarDraft needs to know where to find these.
Also, if you want to view PUD entries, you have to specify a path for a working copy of maindat.war. (it can be changed, in fact you can also specify the same maindat.war you are currently working with, as long as it works fine in WarCraft II)

Tools menu

Simple Image converter

If you want to replace the images in the archives, do it like this:

- Save your image as a 256 color BMP file (uncompressed, or RGB-encoded like some paint programs call it ! **NOT** RLE-encoded !).
- Select *Tools/Image converter*, *Convert BMP to Image Data* and save the resulting file.
- If your picture uses another color palette than the original one, select *Convert BMP to Color Palette* and save it too.
- Now just import the saved image data and replace the original picture in the archive.
- If you use your own palette, import the saved palette over the original one.

Be careful when replacing color palettes; in some cases Warcraft uses the same palette entry for multiple images, so the other images will
look strange if you don't replace them too.

Graphics converter (GfxConv2)

This tool allows you to convert every kind of graphic into a BMP, and vice versa.
WARNING: This is for experts only !
For detailed instructions click here.

Speed editor (Ed278)

With this goodie you can edit the speeds of units in detail. Doesn’t work with WarCraft2 version 1.33 currently.
WARNING: This really is for experts only !
For detailed instructions click here.

Show external PUD

The same as Edit/Show PUD map but it shows an external PUD rather than a PUD-entry from the current archive.

Campaign Builder

Allows you to compile a number of changes to archives into one data file, which can then be distributed along with the Campaign Maker program (WCM.EXE).
For detailed instructions click here.

The Text Editor

To begin with, here are the terms I use in this description:

Text A sub-entry in a text-entry; this can be as short as 1 word or as long as a complete mission briefing.
Line A sequence of characters in a text, which is terminated by <LF> if more characters follow.
Printable Any character that appears in the WarCraft2-fonts; generally this are all letters and digits, as well as some interpunctation marks.
Control-Character Special characters. They are used to specify the color of successive characters, denote line-breaks, etc.

To edit WarCraft 2’s text resources, find the entry you want to change (most entries contain more than 1 text !), and double-click it (or highlight it, and select *Display/Edit Texts* from the Edit menu. The Text-Editor will pop up.

The upper part of the Text-Editor show a list of texts contained in the selected entry. To view or edit one of those simply select it. The text will appear in the edit-area (the lower part of the window).

Important
To get a feeling for how texts are formatted browse through some of the texts first.
Also closely look at the original texts before changing them. Often there are built-in shortcut-keys or special ASCII-codes which are not a known control-character; in most cases (if not always) it’s better to really only edit the text and preserve these.

Control-characters
Editing should be pretty straight-forward, that is, until control-characters are used. Generally, control characters are displayed (and entered) in the following format: `<name>` where `name` is either a number, in which case the ASCII-code with this number will be used, or a description of the control-character. You can view the possible values for name by checking *control char reference*. Using this format for control-characters poses a slight problem however - anything starting with a `<` (opening angle bracket) is regarded as the beginning of a control-character. To enter a real `<` you have to use the special control-character `<ABRK>`. Closing angle brackets can be used normally.

Paragraphs
Some texts display as multiple separated paragraphs in the game, like mission briefings. Each of these paragraphs is entered as one line (use the `<LF>` control-character to separate lines).

Colors
The color of the subsequent characters can be set by using the control-characters `<C0>` and `<C1>` - `<C5>`. `<C0>` sets the color to what it was before the last change. `<C1>` - `<C5>` directly specify a new color to use.

Hotkeys (Keyboard-shortcuts)
Some texts, such as menu-items, include keyboard-shortcuts. The hot key is always the very first character in the text. The text may look like `xExit` or `xE<4>x<1>it` for example (the latter case is typical - the hot key is marked with a different color; but the key WarCraft checks for is always the first). Other texts don’t have hotkeys. Unfortunately there is no way for WarDraft to decide wether or not there is a hotkey - you have to look for these yourself - and if there is one, you **must** also specify one when editing the text.
Note: although the Esc key is used as hotkey in some menu items, it sometimes doesn’t work if you use it in another item.

Unknown control-characters

Sometimes you will notice ASCII-codes (<*number*>); these are control codes where I haven’t yet found out what they are used for. Generally you shouldn’t touch these.

Editing notes

You can toggle word wrap in the edit-area on and off with the word-wrap checkbox. This doesn’t affect the generated text in any way - it’s just for your viewing/editing preferences. Also you can hit *Enter* to add line-breaks or empty lines to get a better overview of long texts - again this doesn’t affect the generated text. If you want to insert a real line-break, use the `<LF>` control character.

WAR file format

Here is a short description in Pseudo-EBNF; all values are hex bytes unless specified otherwise:

```
<WAR-File>        =     <WAR-Header> (8 bytes)
                        <Offset-Table>
                  <Entries>

<WAR-Header>      =     19 00 00 00
                  Number of Entries (WORD)
                        Archive ID (multiple of 1000 (dec.)) (WORD)

<Offset-Table>    =     Offset (LONG)
                        ......            (Number of Entries Offsets totally)
                        (Offset $FFFFFFFF means not available (Demo-Version))

<Entry>           =     <Standard-Entry> | <Pseudo-Entry>

<Pseudo-Entry>    =     anything which is shorter than 4 bytes
                        (calculated from the offsets)

<Standard-Entry>  =     <Entry-Header>
                        Entry-Data  (number of databytes can be calculated
from offsets)

<Entry-Header>    =     Orig.Length and Flags (LONG)
                  (the hi-order-byte is Flags: 20 = compressed, 00 =
uncompressed)
                        (the 3 lower bytes are the uncompressed length
[without header])
```

Knowledge Base format

The knowledge base (default: WAR131.KNW) is a textfile, divided in sections like Windows INI-files; each section represents a different archive.
The very first line in the file has to be "KNWB".
The beginning of a section is a filename enclosed in square brackets, e.g. "[MAINDAT.WAR]" (only the filename, no path information !).
After the section header there are a number of entries, each in the format:

```
      <Entry-Number>=<Type>,[Palette-Number,]<Description>
or    <Entry-Range>=<Type>,[Palette-Number,]<Description>
```

Entry-Number is the number of the entry in the archive (as in WarDrafts *Number*-column).
Entry-Range are two <Entry-Number>s, separated by a "-"
Type is a 3-character description of the entry-format. Currently the following types are supported:

BMP	BMP (Windows-BitMap) picture
CMT	*to enter descriptions for unknown formats*
IMG	Image data file
PAL	Color palette file
PUD	PUD (level) file
STR	String-Table (Text)
WAV	RIFF WAVE Audio file
SMK	Smacker Animation file
XMI	XMI Midi music
GFX	Graphics stored in the same format as units, icons, etc.
CRS	Cursors (mouse pointers)
GFU	Uncompressed graphics (like the symbols for magic and gold,wood,oil,etc.)

Palette-Number is used for Image (IMG), Graphics (GFX), Cursor (CRS) and Uncompressed Graphics (GFU) files only (not optional ! If the palette number is not known use -1)
Description is a short description of the entry
Lines starting with ";" or "#" are comments, blank lines are ignored.
Multiple entries for the same number are allowed, the latest found are used.
Some examples:

```
[MAINDAT.WAR]
192=PUD,PUD Level  1  (Humans)
299=IMG,300,Image: Title screen
300=PAL,Palette: for 299
244-251=PUD,PUD ???
```

Known Bugs

- Viewing an entry as hexdump doesn't work vor very big entries; also preparing the dump is too slow .
- On some configurations Smacker-Anims and audio files don't play (or only once) if WarDraft is installed in a directory with a long filename (> 8 characters)

To do

- Rewrite hexdump routine
- *Online* menu to retrieve the latest knowledge bases via internet
- Speed up the preparation of graphics for the PUD-Viewer
- Avoid flickering when scrolling the map in the PUD-Viewer
- Add more info to the PUD-Viewer (amount of gold in mines, etc.)
- Better integration of GfxConv and Ed278 (e.g. autoloading of selected graphic in GfxConv)
- Make Ed278 compatible with WarCraft2 V1.33

What’s new in this version:

Campaign Builder + Maker: Compile a number of changes to archive files (e.g. PUDs, Graphics, Texts etc.) into one data file for distribution and batch import.

Several bugs fixed.

History

V1.1

- Campaign Builder + Campaign Maker
- Show external PUDs
- Several bugs fixed

V1.0a

- Bugfix for the Text-Editor

V1.0

- Text-Editor for Text/String-Tables
- Minimap for PUD-Viewer
- GfxConv and Ed278 are now included in WarDraft
- Documentation converted to WinHelp format

V0.99

- PUD-Viewer
- Completely rewritten **fast** compression routine
- Supports new graphic types: Cursors, Symbols
- Datafiles can be passed via commandline, so associating .war files with WarDraft is now possible

V0.96

- Can view the unit-, building- and icon graphics.
- Vastly enhanced Knowledge Bases.

V0.95

- Can play XMI midi files (parallel to working with the archive !)
- Export options let you choose standard formats for exports
- Knowledge base entries selectable
- Image converter to add your own pics to the archive
- Some bugs fixed

V0.9

- Faster compression algorithm (written by Daniel Lemberg)
- Can play animation entries (needs SmackPlay)
- Removed some bugs in reading old (V1.2) data archives
- New Knowledge Base for V1.2
- Knowledge Base for V1.31 updated

email: e8900070@stud1.tuwien.ac.at
WWW: http://stud1.tuwien.ac.at/~e8900070/

Suggestions and bug-reports are welcome !

Graphics Converter

Table of contents:

Graphics Converter: How to convert a graphics file to a BMP

- First, export the graphics file and the palette used by it with WarDraft or peace and start GfxConv.

(**IMPORTANT**: use the option "just save it like it is" when exporting the palette !)

- Select the page "Gfx -> BMP"
- Select the type of the graphics file
- Click on "Graphics file" and locate the file you extracted before.
- Click on "Palette file" and locate the extracted palette (this is optional).
- Select how the BMP should look like by entering the respective values in the edit boxes. Choose whatever you like best for editing. (see section *5. Testing animations* for tips)
- Finally click on "Start conversion".

You will be asked for a name for the resulting BMP, as well as for a info file and mask file. You don't have to create the last two (simply click cancel in the open file dialog), but they are necessary if you wish to edit and reconvert the BMP.

After the conversion is done the generated bitmap will be shown.

Graphics Converter: How to reconvert a BMP to a graphics file

- After you have edited the BMP, you have to reconvert it to a graphics file in order to insert it into maindat.war.
- Select the page "BMP -> Gfx"
- Click on "BMP file", "Layout info file" and "Mask file" (optional) to select the respective files.
- Now choose how transparent areas are created:
  (a) If you have a mask file you can use this (I'll explain mask files later).
  (b) You may also choose to make all areas that use a specific color transparent
  (c) Or you don't want to have any transparent areas.
- Click on "Start reconversion".

You will be asked for a name for the resulting graphics file and reconversion will start. After it is done you can import the saved file to maindat.war.

Graphics Converter: Transparent areas

Most graphics contain transparent areas, i.e. where the background shows through. There are two basic methods to create them:

- Reserve a color you don't use anywhere else. Paint the areas that should be transparent in this color and select the same color-# when reconverting the file. This is the **easiest method if you want to create completely new graphics**.

- This method involves making a mask file. Mask files are 2-color (monochrome) BMPs, the same size as the BMP containing the graphics. Where color #1 is used the respective pixel from the graphics BMP will be solid, color #0 makes it transparent. Just have a look at a mask file produced during conversion, and it will be clear.

  Use mask files when you either want to make make **minor changes** to the existing graphics or create completely new ones but need all 256 colors. In the first case, you usually can use the mask file saved during conversion, as long as you don't make any changes to the graphics' outline (for example if you just change some colors). If you change the outline, you will also have to change the mask in the same way.

  If you create completely new graphics, you will have to use a paint program to create a mask as described above. Make sure you save it as a 2-color uncompressed (or RGB-encoded) BMP.

Graphics Converter: Palettes

Finding the correct palette

If you use WarDraft you can find the correct color palette for a graphics file by either looking through the knowledge base with a text editor, or by selecting "Ask for palette always" and then viewing the entry; in the palette dialog the correct palette will be presented as default value.
Note that when exporting the palette you have to use the option "just save it like it is".

Changing the palette

Three words: **DON'T DO IT !** All graphic file belonging to a certain terrain-type use the same palette. So if you change it all the unchanged graphics will look wrong. Well, you could edit them all, but the map-tiles use the same palette too (and there is currently no editor for these).

Color cycling

Warcraft changes certain color numbers dynamically for effects like moving water etc. I currently have no idea which colors are used for cycling and which remain static, so it would be a good idea to use only the colors that are used in the original graphics.
(I created a blue-eyed Eye of Kilrogg; after importing it to maindat.war and starting the game everything worked well; but instead of a blue eye I got one flashing green and red; looked funny though :)

Graphics Converter: Testing animations

- If the images in the graphics file are used in an animation sequence, it is often useful to check *"use X/Y adjustment values in BMP"* when extracting the BMP. This way the images are aligned perfectly.
- For unit graphics always use 5 icons per row.

Speed editor (Ed278)

WARNING:
The Speed editor is currently not working for WarCraft2 version 1.33

Speed Editor: General

Ed278 is an editor for the mini-programs WarCraft II uses to control unit movement, attack and similar events. This programs are stored in in entry #278 of maindat.war, hence the name *Ed278*.
This editor is far from complete; the only thing that's currently working is the speed-editor. But since this is the most important part for most people I decided to release this version anyway.

Ok, some words about these "mini-programs":
All events (like telling a unit to move somewhere or attack something) start one of these mini-programs. The mini-program itself then takes care about moving the unit (pixel-wise), displaying the correct animation frames, playing attack-sound effects, and so on. Often a couple of units use the same mini-program. This is important to know, because changes in the program will affect all these units. If you want to edit only a specific unit of the group, you have to assign a new group for it. See *Grouping* for details.
The mini-programs use a simple instruction set. For the format of entry #278 click here.

Speed Editor: Loading & Saving

There are two ways you can load or save. Either from/to a separate file which was extracted from/is to be inserted to maindat.war, or from/to maindat.war directly. If you want to access maindat.war directly, you have to prepare it first, since *Ed278* cannot handle compressed entries yet.

Here are the necessary steps to prepare maindat.war for Direct Load/Save:

- Extract entry #278 to a temporary file (if WarDraft asks if it shall uncompress it before saving, answer yes).
- Still having selected entry #278 click on *Import*.
- Set *Import file to* the file you just extracted, check *Replace selected entry* and make sure *Compress* is **not** checked.

Speed Editor: The Speed Editor

The Speed Editor is a crude tool that allows you to change the **WAIT** commands of the programs. These commands (there are several different) employ a pause - so usually the more (or longer) **WAIT** commands, the slower the unit behaves.

Select the unit you would like to modify from the *Select program for unit*-list and select the desired action from the *Select event*-list (note that there are events that are used by all or many units; I have found 3 yet, they start at line 1376,   137C and 1830. I would not recommend to change something in these events). The other units affected by the selected program and the program itself are displayed below.
To change the **WAIT** commands, simply select one. You now can use the trackbar to modify the waiting **time** and the radio buttons to change the waiting **type**. (You can also use keyboard shortcuts: 1,2,3,4 set the type, +,- increase/decrease the time).
Do not set the waiting time to zero ! This doesn't work; I think WarCraft II actually interpretes a **WAIT 0** as **WAIT 256**, but I'm not sure. The smallest possible time in a **WAIT** command is **1**.
For more information about waiting types see Appendix A.

A note about speed display in WarCraft II: The **displayed** speed is always relative to the speed of a footman (which is 10). If you change the footman's speed to 5, War2 will still display 10 for the footman and the double of the usual value for the other units. (This only affects the display, not the actual speed the units move.)

Speed Editor: Grouping

WarCraft II usually only has one program for all the units that behave the same. If you edit it, all these units are affected. But often you want to modify just a specific unit, say one of the heroes. This is when you have to go to the *Grouping* page. Here you can move units in and out of groups.

Basic usage:
The top of the window is used to select a group. This is done by selecting any unit that belongs to the desired group in the dropdown-list. You then can put this group into either the left or right list, by pressing *Set source group* or *Set destination group*. Then select the unit you wish to move in the source group (left list) and press *Move unit...* After a short time the unit will belong to the destination group (right list). If you want to have a own group for a unit, simply leave the destination group empty (or press *Clear*).

Many group manipulations change the size of the data; this isn't a problem if you just want to save it and import it to maindat.war later. But if you want to use the much faster way of *Direct Save*, the data size has to be exactly the same as the one currently used in maindat.war for entry #278. That's why there is a checkbox "*Try to keep same file size*". If, after a manipulation, the resulting data is smaller than before, Ed278 can expand it to it's original size, so you can use *Direct Save*. However if the data grows, there is no way but to save normally and import with WarDraft. After this has been done **once**, you can use *Direct Load/Save* again, until some manipulation produces an even bigger file.

- Moving a unit from a group of 2 or more to an already existing group doesn't change data size at all.
- Moving a stand-alone unit to an existing group reduces data size (because 1 group is deleted)
- Creating a new group for a unit increases data size (in most cases).

Speed Editor: Instruction set

As you may have noticed there are still some unknown commands and some that I might have labeled wrong. However, here is what I've found out yet (the code numbers are in hex):

Code 0 - *reset* **- no parameters**
This often does nothing, but it appears in almost every mini-program. When I tried to put it somewhere into a program it often stopped there; so it is definitely not a *no operation* command. I think it is used to set some control variables to defined values.

Code 1 - *wait (type 1)* **- 1 byte-parameter: wait time**
This command halts the unit for the specified amount of time. Mostly used for speed control and animation synchronisation.
There are 4 different types of *wait* commands. I currently don't know the difference; I guess the types affect the ability of the unit to react to attacks or similar things.

Code 2 - *unknown* **- parameters: unknown**
But it exists. It appears in *Event 0* for almost all units (program line $1380).

Code 3 - *goto* **- 1 word-parameter: destination**
This jumps to the specified destination line in the program.Usually this is line $1376, line $137C for finite actions (like walking 32 pixels) or the beginning of the program itself for continuous actions (like attacking).
$1376 and $137C are two do-nothing loops:

```
      1376: reset
            wait1 4
            goto  1376

      137C: goto  137C
```

Code 4 - *show frame* **- 1 byte-parameter: frame number**
Displays the specified animation frame. For units with different angles of view the direction the unit is heading is added internally.
(Directions: N=0 NE=1 E=2 SE=3 S=4; SW,W and NW are just mirrored from SE,E and NE)

Code 5 - *walk* **- 1 byte-parameter: pixels**
Moves the unit the specified amount of pixels along the direction it is heading.

Code 6 - *walk and show* **- 2 byte-parameter: pixels, frame number**
This is just a combination of code 5 and 4.

Code 7 - *wait (type 2)* **- 1 byte-parameter: wait time**
See code 1.

Code 8 - *unknown* **- parameters: unknown**
I haven't seen this yet.

Code 9 - *wait (type 3)* **- 1 byte-parameter: wait time**
See code 1.

Code A - *sound* **- no parameters**
Plays the units "attack"-sound.

Code B - *fire projectile* **- no parameters**
Also non-shooting units do this (they fire an invisible projectile). If it is removed, the unit doesn't do any damage while attacking.

Code C - *wait (type 4)* **- 1 byte-parameter: wait time**
See code 1.

Code D - *sail/fly* **- 1 byte-parameter: pixels**
Same as Code 5, but for ships and flying objects.

Code E - *sail/fly and show* **- 2 byte-parameter: pixels, frame number**
Same as Code 6, but for ships and flying objects.

Format of entry #278

All values are decimal unless specified otherwise:

```
<Entry278>          =     <Header>
                          <Header info>
                    <Program>
                          ...
                          <Program>

<Header>            =     139 x <Offset of program (word)>    Offsets are in the
same order as units in PUDs (127-139 unknown)

<Header info>       =     <Header size (word)>          2*139 = 278
                    <"Normal" units (word)>       110
                          <"Special" units (word)>      29

<Program>           =     <ProgramHeader>
                          <Event-Handler>
                    ...
                          <Event-Handler>

<ProgramHeader>     =     <Offset of Event 0 - Handler> number of Event-Handlers
is usually 7, but not always !
                          <Offset of Event 1 - Handler> Event 1 = Die, 3 = Move,
4 = Attack, others: unknown
                          ...

<Event-Handler>     =     <command>
                          ...

<command>           =     <command code (byte)>
                          [<paramaters>]
```

Campaigns

The basic idea of campaigns was to create a number of PUDs to replace the original ones in MAINDAT.WAR. WarDraft's Campaign Builder allows to compile any number of changes (i.e. entry-replacements) to one or more WAR-archives into one data file. This file can then be distributed with the Campaign Maker (WCM.EXE). WCM reads this data file and automatically applies the necessary changes.

So the term "campaign" refers just to a number of changes to the archives (not necessarily involving PUDs).

Unfortunately there are still restrictions in making "real" campaigns (PUDs that replace the originals), because there is currently no way to edit the mission objectives and unit/building-restrictions. So when you replace an original PUD the objectives and restrictions will be the same as before. For a workaround see the distribution suggestions.
Nevertheless you can use campaigns to apply a number of graphic/sound/text changes. Also carefully designed PUDs, but you have to observe the restrictions mentioned above.

Campaign Builder
Campaign Maker (WCM.EXE)
Distribution suggestions

Campaign Builder

I won't discuss all the menu options since most of them are self-explanatory.

Basics:
WarDraft uses two different file formats for campaigns: template files (.WCT) and data files (.WCD). Template files are the ones you work with while designing your campaign. These are simple text files which can also be edited manually (although I don't recommend that). The Open and Save commands from the File menu use template files.
Data files are for distribution. Once you're content with your campaign you use *Edit/Build...* to create a data file. You only need to distribute this data file and WCM.EXE.
Your campaign can also contain a description. This will be shown when running WCM.

See also: Campaigns overview, Campaign Maker, Distribution Suggestions

Campaign Maker (WCM.EXE)

The Campaign Maker is a small DOS-application that reads a campaign data file (.WCD) and applies the changes specified there.
For instance if your campaign data file is C:\CAMPAIGN\TEST.WCD you type at the DOS-Prompt:

```
WCM C:\CAMPAIGN\TEST.WCD
```

WCM tries to automatically find your WarCraft directory; if that fails for some reason (or you have a second copy of WarCraft installed) you can use the -w option. For example:

```
WCM -w:C:\GAMES\WAR2 C:\CAMPAIGN\TEST.WCD
```

There are also other options available. Just type WCM without any parameters for a short description.

See also: Campaigns overview, Campaign Builder, Distribution Suggestions

Distribution suggestions

Generally the only thing you need to distribute in order for anyone to use your campaign is your campaign data file (.WCD) and a copy of WCM.EXE. Also add a README.TXT that tells the user how to use WCM. Zip these files and upload it to your favorite ftp-server or your WEB-site.
Or create a self-extracting zip (or arj or whetever...) that automatically starts WCM.

If you want to distribute a PUD-campaign, but don't want the restrictions mentioned here, the only workaround (I currently can think of) is: Don't overwrite the original PUDs (i.e. don't use any PUDs in your campaign file) but rather add them to the distribution zip. This way the user has to load the PUDs directly from WarCraft, but you can use an accompanying campaign file for any changes to texts (especially unit-names), graphics, sounds, etc.

If your campaign makes any changes to files that normally are not copied to the harddisk (like MUDDAT.CUD or SNDDAT.WAR) the user has to copy these from the CD into the DATA\ directory of his WarCraft installation dir (same where MAINDAT.WAR etc. resides) **before** running WCM. Future versions of WCM will be smart enough to do that automatically.

See also: Campaigns overview, Campaign Builder, Campaign Maker