What is DiabloTrainer?

DiabloTrainer is a utility that you may use to expand your normal Diablo game. In its basic form, it allows you to save and restore your character-even your multi-player characters!

If you manually activate the "cheat" features, then it also allows you to increase your gold, stats, level, spells, and experience. It also allows you to temporarily own a "SuperWeapon."

DiabloTrainer saves your character as a non-encrypted binary file. This means that you can use a standard hex editor to locate your character's statistics and change them to your liking.

DiabloTrainer will also allow you to link it with a "Plug-In." Plug-Ins are small applications that work with DiabloTrainer to accompolish a hex-editing goal. It can be as simple as a DOS-based program to pump up a particular stat, it can be your favorite hex editor, or it could be a full-blown graphical interface to your equipment. The only limits are those of the programmers out in the world!

How Does DiabloTrainer Work?

DiabloTrainer takes over part of Diablo's memory space. In case you did not know, Diablo saves your network character(s) in your windows directory as DL_INFO*.DRV. These files, approximately 66K each, contain your character's information-encrypted, and probably compressed. Modifying the file is next to impossible. Moving the file to another computer or backing it up to the same computer is also difficult. Many people have searched for a way to break this scheme, as Diablo stores some kind of key in the Windows Registry (I have been told that it stores the file's date/time, your computer's name, your hard drive's label, the geometry of your hard drive, etc).

DiabloTrainer gets around all of that by letting Diablo do all of the work. After all, all of the routines for encryption, validation, compression, etc. already exist in Diablo. Why duplicate the effort? Upon loading a game, Diablo loads your character from disk (unencrypting it along the way). It stores your character in system memory, as plain, non-encrypted, machine-readable data. I take advantage of this and do all saving, restoring, and manipulation directly in data memory. Your programs remain untouched and unpatched, but the data that Diablo keeps in "short term memory" gets "scrambled" to your advantage.

DiabloTrainer takes advantage of a bug (or as Microsoft says "Feature!") available in Windows 95. Normally, Windows will keep each program running in its own virtual memory space. A program is not allowed to touch another's memory, as this can lead to corruption and a crashed system. Fortunately, there is a hole allowing one process to manipulate memory of another process. This manipulation is not direct, but it is direct enough to allow DiabloTrainer to work.

Because this "feature" exists only in Windows 95, DiabloTrainer works only under Windows 95. Presently, it does not work with Windows NT. I have no plans in the near future to add Windows NT compatibility.

Cheating

Many people have asked my opinion on cheating. Many people have flamed me for writing DiabloTrainer. Let's sit down around the fireplace, throw another log into the file, and listen to Uncle Enigma talk about cheating and the history of DiabloTrainer.

DiabloTrainer originally began as a backup/restore utility for network game characters. It would save the DLINFO_?.DRV files that reside in your Windows directory, along with a little snippet of your registry that corresponded with that file. This worked at times, but unfortunately, not all the time and not across machines. It seems that Diablo keeps a record in your system's registry that somehow uniquely tags the save game file. Originally, I was told by several people that Diablo keeps the save file date and time in the registry. I was not able to decode this data so my system merely backed up the save file (keeping its date and time), then the registry entry--it reversed the process to restore the file. This did not always work. Someone on the internet told me that Diablo keeps record of the save game's disk sector. This is a logical hypothes is that explains why the backup/restore utility worked sometime and not others and not across computers. By this time, though, I had already written a memory backup/restore utility and did not have the time to pursue the possiblility of the registry containing disk sector information.

The memory backup/restore software was simple enough and took next to no time to code. The next logical step was to tweak with the memory. I located the position of your stats and your equipment. I decoded the format of a "Gold" inventory slot. This was an easy feature to add. At this point, I planned to stop.

Later that day, several people at work told me that it was possible to use the backup/restore utility to backup and restore OTHER people's characters! It seemed that whoever was in the first character slot (i.e. the creator of the game) could be backed-up/stolen/cloned/ripped, no matter which computer you resided at. This was an interesting fluke... It got to be even more interesting when I added the character selection dialog box (so that you did not have to be creator of the game to use DiabloTrainer).

People ask me: "Why did you do it? You ruined the game!" In all honesty, this program was never meant to be a cheat. It was just for backup and restore, but the cheat routines naturally followed--if I did not do it, someone else would do it: either directly through the game, or by tweaking with my memory-dump files. The cloning feature happened to be a complete accident!

If you are to be playing this game, know who you are playing against! If your friends are cheating, but you do not wish them to cheat, maybe you need new friends. The key words for everything are: SELF CONTROL. Know yourself, know your tendencies, control yourself and any bad tendencies you may posses. Make sure your friends are doing the same. I can exercise self

control, which is why I made the trainer. Except for some experimental characters, I still have my 4th level mage, which I back up to floppy and carry back and forth between work and home.

Everyone using the trainer should know the Golden Rule (no, no, I do not mean "he who has the gold makes the rules!"). If you do not like being fucked with, do not, under any circulstance, fuch with others-no matter how righteous it feels to kill a Pkiller. This rule can be bent a litte when you are joking around with friends that you know extremely well, but when it comes to playing against people you do not know (e.g. Battle.Net), cheating just plain shows a lack of maturity, commitment, experience, and a lust for "the easy way out."

If you cannot handle the responsibility of owning DiabloTrainer, you should not own it.

In all honesty, I have probably spent more time working on the trainer, itself, than actually playing the game! It is a puzzle for me to play with! DiabloTrainer and Diablo's memory space have gotten to be more of a diversion to me than the game itself. I can vouch that Diablo on a LAN is a wonderful experience, when everyone is using the trainer or when no one is using the trainer-but not in a mixed environment, where some are and some are not.

"*If the wrong person preaches a right teaching, even a right teaching becomes wrong. If a right person expounds a wrong teaching, even a wrong teaching becomes right.*" --Muso Kokushi

Frequently Asked Questions
(FAQ)

"Can you put in a feature to duplicate items?"
At the present, no. Maybe someone will discover and release the specifications of items in the dump file? Maybe someone will write a "third-party" tool to mess with that file?
"Your trainer is cool. Can I get the source?"
No. It is not available.
"I was poking with the hex-dump file and discovered that 'this' number means 'this' thing..."
Cool! You can either submit this information to me, which I can distribute or include as a feature in the trainer; or you can write your own program/Plug-In that operates upon the hex-dump file. Or you can do both!
"What did you use to figure out what was in memory?"
Microsoft Developer Studio Professional:
Visual C++ v4.2
Microsoft Developer Network
"What did you write the trainer in?"
See the answer to the previous question
"Can I distribute your DiabloTrainer?"
Sure, but be fair to all parties involved. Do not charge for it. If you are a company that charges "copying fees" for shareware/freeware, make them reasonable. If you are distributing it on the internet, be kind to my ISP. Do not link directly to the archive, download a local copy and put it on your own site (it is relatively small), or put a link to my page.
"Is it shareware or freeware or what?"
Totally free. I don't want anything for it. Well....maybe if you run into me at a club or coffee shop, some caffeine or alcohol would be nice...

Places on the Net

The following internet resources contain useful Diablo information:

http://exo.com/~enigma/diablo.html
Enigma's DiabloTrainer Page
http://www.users.lith.com/~kevin
Kevin's Backup Utility

Places on the net that offer Plug-Ins:
http://www.flash.net/~mage/diablo_uge.zip
The U.G.E. editor for *.DMP files. This is not, exactly, a Plug-In-but it can be used to modify the values in your saved games.

This is not a comprehensive list of Diablo sites. This is a list of resources that would be important to users of DiabloTrainer. If you have a Plug-In or similar utility, drop me a line and I will include it on my site and in this file.

General Operation

DiabloTrainer needs to operate with Diablo running in the background. It does not matter which of the two you start first, but as soon as you click on a button in DiabloTrainer, it searches to verify that Diablo is loaded.

To switch between Diablo, DiabloTrainer, and any other programs you may have running use ALT-TAB. Keep in mind, that when you ALT-TAB out of Diablo, the game is still running in the background, even though you cannot see it. This is fine if you are in town, but when you are in dangerous territory, you could get killed by a monster or another player if you are not quick.

Saving a Character

To save a character, merely start up or join a network game with the character you wish to save. Once you see your character standing around, looking for something to do, Alt-Tab over to DiabloTrainer, then click on the "Save" button. First, DiabloTrainer will ask you which character in the game you are, then it will prompt you for a file name (with a *.DMP extension). Pick a file name to the file to disk. It is approximately 23K in size.

This file may be copied anywhere you wish and restored at a later time.

Restoring a Character

Restoring a character is a little more complex than saving a character. Start up Diablo and select a "dummy" character. This dummy character will be overwritten with the save-game image. It has been reported that you get the best results if the dummy character and the save-game image both have the same name. Create or join a game. Once you see your character standing around, Alt-Tab over to DiabloTrainer. Click "Restore," pick the character slot with your "dummy" character's name, and show DiabloTrainer where the *.DMP file is. The *.DMP memory image file will be restored on top of the present character, replacing all stats and items.

Alt-Tab back over to Diablo. Quite often, this is where people receive a "General Protection Fault" because they did not carefully follow directions. **DO NOT MOVE** your character. Do not allow him to walk around yet. Click on your inventory screen, then disarm and re-arm an item (such as your armor). If you want to be completely safe, it is best, at this point, to hit ESC and quit Diablo or choose "New Game." This will cause Diablo to clean up memory and save your character to its encrypted disk file.

Copying a Character from Single to Multi (And Back Again)

A big "Thank You" goes to Sam Shockey for contributing this information. Be sure to check out his Diablo web site at:
http://www.concentric.net/~shockesj/games/diablo.shtml

How to transfer a single player character to a multiplayer character using the Diablo Trainer Beta 3_4 (Thanks Enigma!).

Load Diablo
Start a single player game with the character you want to transfer
After the game loads press ALT-TAB to switch back to windows
Load the Diablo Trainer
Press Save to save the character to a file
Go Back to the Diablo game (It will be on the taskbar)
Press Esc to get the menu and select New Game
Select cancel to get back to the main menu
Select Multiplayer
Create a New Hero
Select the same class and name of the single player character
Start a Local Area Network (IPX) game (it has also been reported to work with serial-cabe games)
Create the game
Once the game loads press ALT-TAB to switch back to windows
In the Diablo Trainer press restore
Type in the name of the file that you saved the single player to
Hit Ok and answer the prompts that come up
After the file is restored switch back to Diablo
!!!Follow this EXACTLY!!!!
-Press ESC and select New Game (This will save the character) If you try to move the character Diablo will crash, usually without saving the new character.

The following is a modified version, submitted byZachary…

i tested cloned players (people agreeing to be cloned ONLY) from battle.net on my IPX network.. not sure how well it works without IPX support.. all restores were done to the SECOND player slot..

1. Save their player to disk (from multi game) - obvious
2. start a new game on ipx with NEW player, that has the EXACT same name as the person cloned..
3. Empty their inventory
4. close the inventory (so NOTHING is on the screen)
5. ALT-TAB and restore the saved game that you cloned..
6. ALT-TAB back to Diablo
7. Bring up the inventory

8. ALT-TAB to trainer
9. Restore the game AGAIN!
10. hit ESC, NEW GAME..

i have not had a crash under this system yet..

i have had limited success with less than exact name usage (for the new character, sometime capitalization counted, sometimes not)..

If you can't start an IPX network game, back in Windows you have to install the IPX network drivers. Do this in 'control panel', 'Network', and click on 'Add'. The IPX/SPX driver is in the 'Microsoft' group. Consult your Windows manual or online help for detailed information.

Duplicating Items

This was sent to me by "jrl," who found a great way of copying items. There have been some reports that items duplicated this method occasionally disappear. For instance, if the original and copy or two copies end up on the ground at the same time, occasionally both of them will disappear completely from the game! I do have some theories on why this is the case.

i tested cloned players (people agreeing to be cloned ONLY) from battle.net on my IPX network.. not sure how well it works without IPX support.. all restores were done to the SECOND player slot..

1. Save their player to disk (from multi game) - obvious
2. start a new game on ipx with NEW player, that has the EXACT same name as the person cloned..
3. Empty their inventory
4. close the inventory (so NOTHING is on the screen)
5. ALT-TAB and restore the saved game that you cloned..
6. ALT-TAB back to Diablo
7. Bring up the inventory
8. ALT-TAB to trainer
9. Restore the game AGAIN!
10. hit ESC, NEW GAME..

i have not had a crash under this system yet.. this info may be helpful in determining exactly what is going on in regards to saving of the character..

i have had limited success with less than exact name usage (for the new character, sometime capitalization counted, sometimes not)..

Activating the Cheats

In order to use any of the cheats, you must agree to the two disclaimers. Cheating is a very serious offence and should not be taken lightly. It may be acceptable in a LAN or modem game against your friends. It cannot be tolerated over the internet. Battle.Net is a cool thing. It is a free thing. Do not abuse this privilege.

Increase your Gold

This procedure will increase every inventory slot containing gold to the maximum value ($5,000). This can get you a maximum of $200,000 from $40 (assuming you were carrying nothing but a single gold piece in each inventory slot). If all of your inventory slots that contain gold already contain $5,000, you need to right-click on the pile of gold; Diablo will ask you how much to transfer, so type in a number (example: 1), you will then be carrying that many gold pieces, which you can drop into a free inventory slot.

Activate Diablo and the cheat, as described in the general instructions. Once you have your game running and your character standing somewhere, go into the trainer and click on the "Gold" button. The trainer will ask you which character slot yours is in, and will then pump up your gold.

Increase or Decrease your Stats

This procedure will let you specify your "Level-Up" points to any amount you wish between 0 and 255. If you are low level and want points to distribute, which will increase your strength, dexterity, life, mana, etc, then set this to a high number. If you are already a high-level character and are getting annoyed at the constant "Level-Up" button on your game that will not go away because you have no place to distribute your points, then set this value to zero.

Activate Diablo and the cheat, as described in the general instructions. Once you have your game running and your character standing somewhere, go into the trainer and click on the "Stats" button. The trainer will ask you which character slot yours is in, what you want your new "Level-Up" points to be, and will modify your character accordingly.

Increase or Decrease your Level

This procedure will increase your decrease your level. It is most useful for getting into the games and levels you normally would not have access to (e.g. "Hell"). It does ***NOT*** give you any of the benefits of having an increased level-such as "Level-Up" points or increased stats or increased experience. It just changes the number associated with your character's level.

Activate Diablo and the cheat, as described in the general instructions. Once you have your game running and your character standing somewhere, go into the trainer and click on the "Level" button. The trainer will ask you which character slot yours is in, what you want your new level to be, and will modify your character accordingly.

Increase your Spells

This procedure will increase all of your available spells to their natural maximum of 15. You will notice that there are several blanks in your character's spell book. This is unavoidable at this time. These spells are unique spells that cannot be directly learned by your character, but can be attached to unique items…something tells me that someone is going to crack the *.DMP codes to give you these spells!

Activate Diablo and the cheat, as described in the general instructions. Once you have your game running and your character standing somewhere, go into the trainer and click on the "Spells" button. The trainer will ask you which character slot yours is in and will modify your character accordingly.

Increase your Experience

This procedure will increase your experience points to the required value for the next level. All you need to do is kill something tiny, and you will go up a level.

Activate Diablo and the cheat, as described in the general instructions. Once you have your game running and your character standing somewhere, go into the trainer and click on the "Experience" button. The trainer will ask you which character slot yours is in and will modify your character accordingly.

The SuperWeapon!

This procedure will change your current weapon into a "SuperWeapon." It is a good idea to use DiabloTrainer's backup feature beforehand, as the SuperWeapon feature is still experimental. Select a name and a graphic for your weapon. This information will be temporarily written over the item your character is carrying in his right hand (the weapon box on the LEFT side of the screen when you look at your inventory). A name and graphic are randomly generated each time you start the trainer, but you may use any you wish. Someone named "crymson" submitted a list of graphics codes.

Keep in mind that the SuperWeapon is only a temporary item. You can drop it and pick it up, but if you leave it sitting around on the ground for too long, give it to another character, start a new game, or quit Diablo, then it will revert back to the item it originally was.

hku@slip.net

How Plug-Ins Operate

A "Plug-In" is a "third-party" program. They are simple, easy interfaces between Diablo's memory space and a program of your choosing. You may associate one of the five buttons at the bottom of the screen with a program. Once you click on the button, DiabloTrainer will save the game in progress to a temporary file, load up the associated program (using the temporary file path/name on the command line), monitor until the associated program has exited, and then restore the temporary file back into memory. This allows you to quickly link up your favorite DOS or Widows program to a button-such as your favorite hex editor, or a plug-in you discovered on the net.

At the time of this writing, the only Plug-In that I know of is on the DiabloTrainer home page. It was written by me, Enigma, and includes full Visual Basic source code. It simply changes your character's name. This source code can be copied and modified to your liking. If you discover some "feature" in the hex *.DMP files that you wish to exploit, write a Plug-In for it!

If you forget what programs you have associated with which buttons, simply move the mouse over a Plug-In button for a few seconds. A "Tool-Tip" will appear, telling you what the name of the associated program is.

How to Change or Add a Plug-In

To associate a program (Plug-In) with a button, simple hold down the shift key while clicking on the button with the mouse. A dialog box will appear, asking you for a file name. Locate the Plug-In program and tell the file selector dialog to "open" the file. Then DiabloTrainer will ask you to describe the program. This description is what appears in the "Tool-Tip" text that floats above the Plug-In button.

How to Create a Plug-In

In order to create a Plug-In, you need to know a programming language that will allow you to create an EXE or COM program. Examples are: Visual C++, Visual Basic, QuickBasic (but *NOT* Qbasic!), Pascal, etc.

Your program should operate upon a standard "*.DMP" file. The complete name and location of the *.DMP file will be passed to your software on the command line. So, for example, if the *.DMP file is called TEMP.DMP and is located in the C:\TEMP directory and your program is called HEXMASTER.EXE, then the following command line is, effectively, used:
HEXMASTER.EXE C:\TEMP\TEMP.DMP

In case the format of the *.DMP files changes in the future, be aware that there is a 21-byte header at the beginning of the file: the name of the file, a CR (chr(13)), a LF (chr(10)), the number 1 (for version 1), an EOF marker (chr(26)), and a zero. It would be a good idea to, at least, check for the version numer, if not the whole block.

A sample Plug-In can be located on the DiabloTrainer web site. This Plug-In allows you to change your character's name. It was written in Visual Basic, and contains complete source code. Use or modify this program to your liking.

Inventory Codes

As yet, I have no solid, verifiable information. Crymson has submitted some useful information about weapon attributes and graphic codes.

Allow me to take a second to present my theories on how Diablo deals with weapons. Presently, this is just an untested hypothesis....

If you have tweaked with your items in the hex dump of your character, you have probably found where Diablo stores the information about those items statistics (bonuses and such). You have also probably tried to modify them. You have probably noticed (through your playing, or through testing the SuperWeapon) that those stats do not stay permanent. You may have also tried cloning some of your items...load a game, drop some items, re-load the game, drop them again, repeated until you have enough items. This doe not always work, as certain duplicated items, when dropped on the ground with their original counterparts, disappear from the game without a trace.

My theory belives that Diablo has a centralized item database residing somewhere on your system, probably the Diablo CD. This database is the same for everyone. When you load and save a character through Diablo, the individual items and their statistics are not saved, but rather a REFERENCE to the item's position in the item database. You would think, "why don't they use the database for everything, then?" The answer is kind of simple: You may have an "8x" CD-ROM drive, but your buddy may have a "2X" drive, which would slow down his game tremendously. Diablo fixes this by copying the item's entry out of the database and into main memory along with a pointer to where in the database the item is located. Because it is in memory, it can be instantly accessed. This copy, though, is only a working copy. Because all of the item's details are stored in the database, there is no point in storing the redundant information in your save game, so it only stores the location in the item database (effectively loosing all of your modifications in memory).

When it comes to dup'ed items disappearing, there is an equally simple plausible explanation. I believe that every time a new item appears (such as in the dungeon or in the store), a serial number or GUID is associated with the item. The serial number is randomly generaed-probably based by the time and some characteristics of your computer (because with the number of possible character interactions Battle.Net, you need more variables than just time). When you pick up an item, it has this GUID. When you save your game, the same GUID is saved with it. If you duplicate the item using DiabloTrainer's Save/Restore feature, all duplicates of the item will have the exact same GUID. If Diablo ever detects two identical items with two identical GUID's it removes them from the game-obviously one, or both, are copies. The key, then, is to duplicate an item, locate the GUID, and change it.

Comments and suggestions would be greatly appreciated. I have very little time to verify this theory.

Inventory Graphic Codes

The following information was submitted to me by crymson. crymson, thank you very much! It looks like you put quite a lot of time and effort into this list. I am sure that it will be of value. It was updated by CuitX.

Diablo's Items Informations

This is given by CutiX.
Initial work by crymson (hku@slip.net)

------ Items Placement in file -----------

Offset of 1st item: 780 => Head
2nd => Left Ring
3rd => Right Ring
4th => Amulet
5th => Weapon
6th => Shield
7th => Armor

Full Structure size 368 bytes

------ Items Structure -----------

Unidentified Name: 64 bytes (null terminated)
Full Name: 64 bytes (null terminated)

Equip Placement: 1 byte
01 - One Handed Weapon
02 - Two Handed Weapon
03 - Body Armor
04 - Head Gear
05 - Ring
06 - Amulet
07 - Non-Equip
Classifies: 1 byte
01 - Weapon (Displays Damage)
02 - Armor (Displays AC)
03 - No Display
Skip 1 byte
Item Graphic (see list down): 4 bytes
Item's unidentified price: 4 bytes
Item's retail price: 4 bytes
Minimum Damage Done: 4 bytes
Maximum Damage Done: 4 bytes
Armor AC: 4 bytes
Skip 1 byte
Magic Special Ability: 1 byte
20: Lightning: 2-20 Additional Damage
40: Vampire:Steals Mana
Combat Special Ability: 1 byte

- 01:Blood: Steals 5% Life
- 04:Swiftness: Fast Attack
- 08:Speed: Faster Attack
- 10:Haste: Fastest Attack
- 20:Balance: Fast Hit Recovery
- 80:Harmony: Fastest Hit Recovery

Additional Damage: 1 byte

- 02:Shock: Lightning Arrow 1-6 damage
- 04:Thorns: Attacker gets 1-3 damage

Magic Item: 4 bytes (P=Potion, S=Scroll, E=Elixir)

- 02:PFHealing
- 03:PHealing
- 06:PMana
- 07:PFMana
- 0A:EStrenght
- 0B:EMagic
- 0C:EDexterity
- 12:PRejuvenation
- 13:PFRejuvenation
- 15:SApocalypse
  - SHealing
  - SIdentify
  - SInfravision
  - SNova
  - SPhasing
  - STeleport
  - STownPortal
- 16:SChainLightning
  - SFireBall
  - SFireWall
  - SFlash
  - SGolem
  - SGuardian
  - SInferno
  - SLightning
  - SResurect
  - SStoneCurse
- 17:Staff
- 18:Book
- 19:Ring
- 2C:ESpectral

Spell: 4 bytes

- 01:FireBolt
- 02:Healing
- 03:Lightning
- 04:Flash
- 05:Identify
- 06:FireWall
- 07:TownPortal
- 08:StoneCurse
- 09:Infravision
- 0A:Phasing
- 0B:ManaShield
- 0C:FireBall
- 0D:Guardian

0E:ChainLightning
0F:FlameWave
12:Nova
14:Inferno
15:Golem
17:Teleport
18:Apocalypse
1E:Charged Bolt
1F:Holy Bolt
20:Resurect
21:Telekinesis
Current Charges: 4 bytes
Max Charges: 4 bytes
Current Durability: 4 bytes (FF if indestructible)
Maximum Durability: 4 bytes (FF if indestructible)
Damage Bonus *: 4 bytes
To Hit Bonus: 4 bytes
Armor Bonus - Percentage: 4 bytes
Strength: 4 bytes
Magic Skill: 4 bytes
Dexterity: 4 bytes
Vitality: 4 bytes
Fire Resistance - Percentage: 4 bytes
Lightning Resistance - Percentage: 4 bytes
Magic Resistance - Percentage: 4 bytes
Magic Points *: 4 bytes
Health Points *: 4 bytes
Damage Bonus (No Blue Highlight): 4 bytes (Gore/Slaying)
Absorption Mask & Angel Spell Increase: 4 bytes
For Deflexion/Absoption items
FD FF FF FF gives -3
FC FF FF FF gives -4
For Angel/Arch-Angel
01 Gives 1 Level
02 Gives 2 Levels
Skip 8 bytes
?: 4 bytes
44 Always ?
Skip 4 bytes
Minimum Additional Damage: 4 bytes (Lightning/Shock/Thorns)
Maximum Additional Damage: 4 bytes
Skip 8 bytes
Prefix of Item: 1 byte
00 Meteoric/Mithril/Platinium: +x% to Hit
02 Brutal/Massive/Savage: +x% Damage
04 Fine/Lord/Master/Soldier: +x% to Hit, +y% Damage
06 Awesome/Blessed/Glorious/Saintly: +x% to AC
08 Crimson/Garnet/Ruby: Resist Fire +x%
09 Azure/Blue/Cobalt/Lapis/Sapphire: Resist Lightning +x%
0A Aearl/Crystal/Diamond/Ivory: Resist Magic x%
0B Amber/Emerald/Jade/Obsidian/Topaz: Resist All x%
0E Angel/Arch-Angel: Increase Spells by 1/2 level
0F Bountiful/Plentiful: Extra Charges
11 Lightning: Extra Lightning Damage
21 Drake/Raven/Snake: +x Mana

FF None
Suffix of Item: 1 byte
13 Giant/Titan: +x Strength
15 Sorcery/Wizardry: +x Magic
17 Accuracy/Precision/Skill/Perfection: +x Agility
19 Life/Vigor: +x to Vitality
1B Heavens/Moon/Stars: +x to all
1E Absorption/Deflexion: -x Damage
1F Tiger/Wolf/Lion: +x to HP
23 Structure: Increased Durability
24 ?: Decreased Durability
25 Ages: Item is indestructible
26 ?: Displays Light Radius
2B Shock: 1-6 Lightning Damage
39 Piercing/Bashing: Damage Target's Armor
37 Vampire: Steals 5% Mana
38 Blood: Steals 5% Life
3A Speed/Haste/Swiftness: Faster Attacks
3B Harmony: Fastest Hit Recovery
3D Gore/Slaying: +x Damage
FF None
Skip 18 bytes
Strength requirement: 1 byte
Magic requirement: 1 byte
Dexterity requirement: 1 byte

* Hundreds Incrementation - Increments by 100, for example:
64 raises it one point, C8 raises it two..

GRAPHIC LIST

01 White Scroll
02 Brown Scroll
03 Blue Scroll
04 Small Gold Pile
05 Medium Gold Pile
06 Large Gold Pile
07 Thick Ring
08 Ruby Ring
09 Thorny Ring
0A Sapphire Ring
0B Skull Ring
0C Metal Ring
0D Ebony Gold Ring
0E Orange Gem Gold Ring
0F Thin Rainbow Vial
10 Large Rainbow Vial
11 Ruby Vial
12 Steel Blue Diamond Ring
13 Player Ear w/ Steel Earring
14 Player Ear w/o Earring
15 Player Ear w/ Gold Earring

16 Hellraiser Orb
17 Hellraiser Box
18 Hellraiser Pyramid
19 Large Ruby
1A Red Christmas Tree Ornament
1B Blue Sapphire Cube
1C Orange Pyramid
1D Thin Black Vial
1E Large Black Vial
1F Semen Jar
20 Red Health Potion
21 Full Rejuvanation Flask
22 Orange Potion
23 Full Health Potion
24 Thick Black Potion
25 Rejuvenation Flask
26 Large Semen Jar
27 Magic Elixir
28 Brain
29 Claw
2A Tooth(?)
2B Brush(?)
2C Blue Ivory Amulet
2D Standard Amulet
2E Black Eye Amulet
2F Bug Brooch
30 Ankh
31 Pouch
32 Bastard Dagger
33 Red Handle Dagger
34 A Bottle o' Brew
35 Jagged Dagger
36 Black Handle Dagger
37 Snap Blade
38 Long Sword
39 Great Sword
3A Jagged Sword
3B Mace
3C Red Handle Sword
3D Emperor's Great Sword
3E Curved Blade
3F Spiked Mace
40 Short Sword
41 Claymore
42 Wooden Club
43 Sabre
44 Legion Sword
45 Bone Club
46 Spiked Club
47 A Board With A Nail
48 Pirate Sword
49 Blood Angel Sword
4A Pain Sword
4B Helm
4C Rock

4D Spiked Crown
4E Thanatos' Crown
4F Leorics Helm
50 Jester Cap
51 Jeweled Helm
52 Goggled Helm
53 Buckler
54 Horned Helm
55 Great War Helm
56 Black Book
57 Red Book
58 Blue Book
59 Black 'Shroom
5A Skull Cap
5B Cap
5C Mined Gold
5D Cushioned Skull Cap
5E Red Tunic
5F Gold Crown of Death
60 Map/Diagram
61 Tome of Knowledge
62 Roman Legion Helm
63 Samurai Helm
64 Great Shield of Lightning
65 Double Bladed Axe
66 Bow
67 Black Dragon Plate
68 Hack Axe
69 Round Shield
6A Butcher's Cleaver
6B Studded Black Leather Armor
6C Bone Bow
6D Red Striped Staff
6E Blood War Sword
6F Chain Mail Dress
70 Throwing Axe
71 Coat of Arms Shield
72 Chain Mail
73 Thick Shield
74 Grinning Death Shield
75 Devil's Shield
76 Hunter's Bow
77 Ugly One-Horned Mule Bow
78 Fish Bow
79 Hammer
7A Steel Hammer
7B Ruby Staff
7C Hooked Spear
7D Shaman's Staff
7E Tavern Sign
7F Leather Roman Legion Armor
80 Rags
81 Pajama Armor
82 Three Ball Chain Whip
83 Morning Star

84 Tower Shield
85 Warrior's Bow
86 Katana
87 Cloth Tunic
88 Samurai Armor
89 White Robes
8A Dark Mage Robes
8B Leather Shirt
8C Anvil
8D Dark Axe
8E Halberd
8F Death Axe
90 Hatchet
91 (No Fucking Idea)
92 Lion Shield
93 Eagle Shield
94 Black Lion Shield
95 Magi Robes
96 Lava Robes
97 Knight's Armor
98 Horned Black Dragon Armor
99 BreastPlate
9A Studded Robes
9B Skull Staff
9C Ripper Axe
9D Padded Plate
9E Crossbow
9F Horned Dragon Plate Armor
A0 Runed Cleaver
A1 Ivory Katana
A2 Spiked Baseball Bat
A3 Dwarves Double-Bladed Axe
A4 Bow of the Wind
A5 Composite Bow
A6 Reaper Spear
A7 Stone Bow
(167 Item Graphics)
--
crymson

DLINFO_?.DRV Files

Although DiabloTrainer gets around directly using the C:\WINDOWS\ DLINFO_?.DRV files, the following information was anonymously submitted to me about the steps in Diablo's Character file read-in process:

*Opens DLINFO_0.DRV
*Closes it
*Gets File Attributes +H +S +A - Checks to see if Attr are correct
*Ioctl: SubF 08
Fndfirst - Volume Label - Checks for volume label
Loctl: SubF 0D
Query
DiskInfo: bytes/sector - Gets info of HD I presume
Total number of allocation units
Number of sectors/allocation
Number of free allocation units

More Inventory Codes

The following information was submitted to me by an intelligent fellow who goes by the name of "Icemaker":

When creating a new item for use in Diablo, it is easiest to start with an item of the same type. If you want to create a ring, use an existing ring as the template. If you want to create a staff, use an existing staff as the template. This is very important as we do not have a complete mapping as to each value that an item requires. Using an existing item as a template ensures that most (if not all) of the required values will be filled in properly.

Search the .DMP file for the item that you are carrying that you wish to use as a template, and when you have found it, back up by 64 bytes. This is the actual start of the item. In the text below, I am giving hex offset numbers to assist in locating the appropriate values. Since the location of the item in your DMP file may differ slightly, simply add the offset value to your beginning value.

This is by no means a comprehensive list of values. There is still a lot of information stored with an item that we do not understand. I recently modified a staff so that a fighter could use it: I renamed it and lowered the magic requirement. When Adria recharged it, the staff reverted to the original name and magic requirement (it may be that there are some checksum values included in the system).

Note: Some of the lists of byte values came from equipment.txt included with the Diablo Trainer Beta 5.3. Special thanks to crymson (hku@slip.net) for compiling this.

HEX
OFFSET LENGTH DESC.

00 63 bytes? Short ("unidentified") name of the item. If you had encountered it in the dungeon, this is what you would have seen. This is a null-terminated string.

40 63 bytes? Long ("Identified") name of the item. This is a null-terminated string.

80 1 byte Identifier as to the equipment placement:

Equipment Placement:
01 - One Handed Weapon
02 - Two Handed Weapon
03 - Body Armor
04 - Head Gear
05 - Ring
06 - Amulet
07 - Non-Equip

Note that making a two handed weapon (such as a stave) into a one-handed weapon so that you can use a shield will cause Diablo to crash when you attempt to equip the shield. From what I can tell, Diablo attempts to load graphics of your character holding the staff and using a shield...and cannot find them. This may work with swords...There is prob'ly another value to tell Diablo what your character should look like when Equipped with this item...I just don't know what it is.

81 1 byte Identifier as to the classification of the item:

Classification:
01 - Weapon (Displays Damage)
02 - Armor (Displays AC)
03 - No Display

83 1 byte Graphic display. See Table B below.

87	2 bytes	Unknown
8b	4 bytes	Full price for the item at Griswald's/Adria's (enter in reverse byte order)
8f	2 bytes	Minimum Damage
93	2 bytes	Maximum Damage
97	1 byte?	Armor rating
a3	1 byte?	Spell that the staff holds. See Table A below.
a7	1 byte?	Current number of charges (applies to staves)
ab	1 byte?	Total number of charges possible (applies to staves)
b7	1 byte?	+ x% to damage
bb	1 byte?	+ x% to hit
bf	1 byte?	+ x% to armor rating
c3	1 byte?	+ x Strength
c7	1 byte?	+ x Magic
cb	1 byte?	+ x Dexterity
cf	1 byte?	+ x Vitality
d3	1 byte?	Resist fire x% (max of 75%)
d7	1 byte?	Resist lightning x% (max of 75%)
db	1 byte?	Resting magic x% (max of 75%)
e7	1 byte?	+ x to damage
123	1 byte?	Strength requirement to use this item
124	1 byte?	Magic requirement to use this item
125	1 byte?	Dexterity requirement to use this item

Table A. Staff Spells

00 = No spell
01 = Firebolt
02 = Healing
03 = Lightning
04 = Flash
05 = Identify
06 = Fire Wall
07 = Town Portal
08 = Stone Curse
09 = Infravision
0a = Phasing
0b = Mana Shield
0c = Fireball
0d = Guardian
0e = Chain Lightning
0f = Flame Wave
10 = Doom Serpents
11 = Blood Ritual
12 = Nova
13 = Invisibility

14 = Inferno
15 = Golem
16 = Blood Boil
17 = Teleport
18 = Apocalypse
19 = Etherealize
1a = Item Repair
1b = Staff Recharge
1c = Trap Disarm
1d = Elemental
1e = Charged Bolt
1f = Holy Bolt
20 = Resurrect
21 = Telekinesis
22 = Heal Other
23 = Blood Star
24 = Bone Spirit
25 = Crash System (this will GPF Diablo when you Equip the staff).

Table B. Item Graphics

01 White Scroll
02 Brown Scroll
03 Blue Scroll
04 Small Gold Pile
05 Medium Gold Pile
06 Large Gold Pile
07 Thick Ring
08 Ruby Ring
09 Thorny Ring
0A Sapphire Ring
0B Skull Ring
0C Metal Ring
0D Ebony Gold Ring
0E Orange Gem Gold Ring
0F Thin Rainbow Vial
10 Large Rainbow Vial
11 Ruby Vial
12 Steel Blue Diamond Ring
13 Player Ear w/ Steel Earring
14 Player Ear w/o Earring
15 Player Ear w/ Gold Earring
16 Hellraiser Orb
17 Hellraiser Box
18 Hellraiser Pyramid
19 Large Ruby
1A Red Christmas Tree Ornament
1B Blue Sapphire Cube
1C Orange Pyramid
1D Thin Black Vial
1E Large Black Vial
1F Semen Jar
20 Red Health Potion
21 Full Rejuvanation Flask
22 Orange Potion

23 Full Health Potion
24 Thick Black Potion
25 Rejuvenation Flask
26 Large Semen Jar
27 Magic Elixir
28 Brain
29 Claw
2A Tooth(?)
2B Brush(?)
2C Blue Ivory Amulet
2D Standard Amulet
2E Black Eye Amulet
2F Bug Brooch
30 Ankh
31 Pouch
32 Bastard Dagger
33 Red Handle Dagger
34 A Bottle o' Brew
35 Jagged Dagger
36 Black Handle Dagger
37 Snap Blade
38 Long Sword
39 Great Sword
3A Jagged Sword
3B Mace
3C Red Handle Sword
3D Emperor's Great Sword
3E Curved Blade
3F Spiked Mace
40 Short Sword
41 Claymore
42 Wooden Club
43 Sabre
44 Legion Sword
45 Bone Club
46 Spiked Club
47 A Board With A Nail
48 Pirate Sword
49 Blood Angel Sword
4A Pain Sword
4B Helm
4C Rock
4D Spiked Crown
4E Thanatos' Crown
4F Leorics Helm
50 Jester Cap
51 Jeweled Helm
52 Goggled Helm
53 Buckler
54 Horned Helm
55 Great War Helm
56 Black Book
57 Red Book
58 Blue Book
59 Black 'Shroom

5A Skull Cap
5B Cap
5C Mined Gold
5D Cushioned Skull Cap
5E Red Tunic
5F Gold Crown of Death
60 Map/Diagram
61 Tome of Knowledge
62 Roman Legion Helm
63 Samurai Helm
64 Great Shield of Lightning
65 Double Bladed Axe
66 Bow
67 Black Dragon Plate
68 Hack Axe
69 Round Shield
6A Butcher's Cleaver
6B Studded Black Leather Armor
6C Bone Bow
6D Red Striped Staff
6E Blood War Sword
6F Chain Mail Dress
70 Throwing Axe
71 Coat of Arms Shield
72 Chain Mail
73 Thick Shield
74 Grinning Death Shield
75 Devil's Shield
76 Hunter's Bow
77 Ugly One-Horned Mule Bow
78 Fish Bow
79 Hammer
7A Steel Hammer
7B Ruby Staff
7C Hooked Spear
7D Shaman's Staff
7E Tavern Sign
7F Leather Roman Legion Armor
80 Rags
81 Pajama Armor
82 Three Ball Chain Whip
83 Morning Star
84 Tower Shield
85 Warrior's Bow
86 Katana
87 Cloth Tunic
88 Samurai Armor
89 White Robes
8A Dark Mage Robes
8B Leather Shirt
8C Anvil
8D Dark Axe
8E Halberd
8F Death Axe
90 Hatchet

91 (No Fucking Idea)
92 Lion Shield
93 Eagle Shield
94 Black Lion Shield
95 Magi Robes
96 Lava Robes
97 Knight's Armor
98 Horned Black Dragon Armor
99 BreastPlate
9A Studded Robes
9B Skull Staff
9C Ripper Axe
9D Padded Plate
9E Crossbow
9F Horned Dragon Plate Armor
A0 Runed Cleaver
A1 Ivory Katana
A2 Spiked Baseball Bat
A3 Dwarves Double-Bladed Axe
A4 Bow of the Wind
A5 Composite Bow
A6 Reaper Spear
A7 Stone Bow

(167 Item Graphics)
--
crymson

More Inventory Codes...

This information was sent to me by Martigan:

It looks like information in each item record starts off with the basic name of the item. Like a KING'S SWORD OF CARNAGE would have a basic name of Bastard Sword (if that is what it is). I will use the beginning of that field as my offset reference--also offsets are in decimal--values are hex.

Offset	Length	WHAT IS IT?
0	64	Basic name of item ("Bastard Sword") (terminated by x00)
64	64	Special name of item ("King's sword of carnage")
128	1	Type of item: 01: One-hand 02: Two-hand 03: Body (armor) 04: Head (helmets and such) 05: Finger (rings) 06: Neck (amulets) 07: Un-equipable (gold)
129	1	Display 01: Weapon (display damage) 02: Armor (display armor value) 03: No display
130	1	??
131	1	Graphics ##: Refer to crymson's list below.
132	3	??
135	4	Base item price (ex: what a bastard sword would cost)
139	4	Modified item price (ex: what a King's sword of carnage would cost) (zeroed if only a basic item)
143	4	Minimum base damage (weapons only)
147	4	Maximum base damage
151	4	Base armor class
155	4	??
159	4	??
163	4	??
167	4	??
171	4	??
175	4	Current durability
179	4	Maximum durability
183	4	+% Damage modifier (x01 equals +1%, x02 equals +2%, etc.)
187	4	+% To-hit modifier (same as above)
191	4	?? (armor modifier??)
195	4	Strength modifier
199	4	Magic modifier
203	4	Dexterity modifier
207	4	Vitality modifier
211	4	+% Magic resistance modifier (same as damage mod above)
215	4	+% Fire resistance modifier
219	4	+% Lightning resistance modifier
223	48	?? (probably several 4-length fields)
271	1	What to display in item description (prefix?)

		04: Show armor bonus 06: Show armor bonus (same as above???) 0B: Show all resistances +% (jade, obsidian, etc.)
272	1	What to display in item description (suffix?) 13: Show strength bonus 15: Show magic bonus 19: Show vitality bonus 1B: Show all attributes bonus
273	18	?? (probably several different fields)
291	4	Strength needed to wield
295	1	This byte is always x01 in every item I have seen.
296	11	??
307	5	These bytes seem to always differ from item to item. Possibly a checksum or item serial number? Could be the reason why items do not stay permanent--Diablo has them registered or something.
312	to end	The rest of the bytes in each item record I do not know. Maybe thinking some data on where an item was found, or toggles for unique items, or something.

More Inventory Information

This was sent to me by Mandragoran ...

Diablo Save Game File

Character Stats:

(offsets here are from start of file)

Offset Description

0567h - Name, null terminated

05A8h - Strength Base (LONGINT)
05ACh - Strength Now (LONGINT)

05B0h - Magic Base(LONGINT)
05B4h - Magic Now (LONGINT)

05B8h - Dexterity Base(LONGINT)
05BCh - Dexterity Now (LONGINT)

05C0h - Vitality Base(LONGINT)
05C4h - Vitality Now (LONGINT)
05C8h - points left to distribute
05CCh - bonus to % to hit (or dexterity bonus....)
05D0h - damage bonus (doesn't stay after inventory is moved)

0606h - Experience (LONGINT)
0614h - Gold Pieces (LONGINT)

063Ch - Random number seed for blacksmithe/towne(LONGINT)
0640h - Random number seed for dungeon level 1 (LONGINT)

09E8h - Equipped Armor Type! (Think this is wrong).

00/014 - 01h,00h
00/040 - 72h,AAh
00/044 - 00h,01h
00/092 - 03h,02h
00/096 - 00h,02h
00/101 - 6Fh,A8h
00/357 - 70h,A9h
01/101 - 71h,A6h
01/357 - 72h,A7h
02/192 - 06h,08h
02/464 - 05h,01h
04/488 - FFh,06h
04/489 - FFh,00h
04/490 - FFh,00h
04/491 - FFh,00h
13/240 - 01h,00h
14/364 - 06h,00h
14/368 - 34h,3Ah

14/372 - 51h,33h

Item Stats:

(Offsets here are within each item record. If you find the Normal Name in each item record and count back 35h bytes, you'll be at the start of that item's data)

First item after player is head equip slot
Second item is chest equip slot
Third item is unknown or unused
Fourth item is right hand (left side) weapon
Fifth item is left hand (right side) weapon
Other items seem to be inventory slots.

0000h - Equip Status (FFFFFFFFh = not equipped, other = equipped)

0034h - Identify status - 0=non-magic, 1=unknown magic, 2=unknown unique,
05

0035h - Normal Name
0075h - Magic Name

00B5h - Item Equip Slot, 04 -head, 03 - chest, 02 -, 01 - hands (BYTE)
00B6h - Item Type?? 01 - weapon, 02 - armor (BYTE)
00B7h - Item picture number - might mean more than that :)

00C0h - Damage Min (LONGINT)
00C4h - Damage Max (LONGINT)

00C8h - Armor Class, higher is better (LONGINT)

00E0h - Durability Now (LONGINT)
00E4h - Durability Base (LONGINT)

00E8h - + damage for magical items
00ECh - + to hit % for magical items
00F0h - + armor class for magical items
00F4h - + str for magical items
00F8h - + magic attribute for magical items
00FCh - + dexterity for magical items
0100h - + vitality for magical items
0104h - resist fire % for magical items
0108h - resist lightning % for magical items
010Ch - resist magic % for magical items

0127h - message # to display when identifying - FFh=no message
00 - to hit: +?
01 - to hit: +?
02 - damage: +?
03 - damage: +?
04 - damage: +? to hit: +?
05 - damage: +? to hit: +?
06 - armor class

07 - armor class: +?
08 - resist fire
09 - resist lightning
0A - resist general magic
0B - resist all
0C - spell cost: -?%
0D -
0E

0129h - str required to wield/number of unique item

Item Picture Numbers:
---

01 - scroll
02 - odd looking thing
03 - gold ring
04 - ruby ring
05 - 3 coins
06 - 6 coins
07 - more coins
08 - half-filled red potion
09 - half-filled yellow potion
0A - 3 ruby ring
0B - vine(?) ring
0C - blue-twined ring
0D - copper ring
0E - silver ring
0F - 6 ruby ring
10 - copper, upside-down triangle amulet
11 - red gem
12 - gold key
13 - green key
14 - silver key
15 - silver chalice
16 - gold sphere
17 - obsidian cube
18 - ivory tetrahedron
19 - small bag
1A - saddlebag/backpack
1B - bread (???)
1C - steak
1D - orange potion
1E - half-filled orange (red???) potion
1F - half-filled black potion
20 - half-filled yellow potion
21 - half-filled white potion
22 - blue potion
23 - serpentine dagger
24 - straight dagger
25 - blue hilted longsword
26 - ruby hilted longsword (very cool looking)
27 - funky bladed sword f-something
28 - mace
29 - red hilted longsword
2A - wicked looking black-hilted sword

2B - scimitar
2C - cool mace
2D - standard shortsword
2E - claymore
2F - club
30 - cutlass
31 - broadsword
32 - metal helm with horizontal eye-slit
33 - black rock
34 - spiked collar
35 - blue-gemmed crown
36 - crown
37 - a mask
38 - evil mask
39 - metal helm, face exposed
3A - normal round buckler
3B - horned helm
3C - winged helm
3D - orange book
3E - orange book
3F - blue book
40 - large bag
41 - skull cap
42 - normal cap
43 - cool blue shield with beast upon it
44 - double-bladed axe
45 - longbow
46 - breastplate
47 - halberd
48 - round wooden shield
49 - the Butcher's Cleaver
4A - studded leather armor
4B - spiked bow
4C - staff
4D - black two-handed longsword
4E - ring or chainmail
4F - small axe
50 - blue and yellow shield
51 - scale mail
52 - metal shield with cross
53 - a pick? some sort of gardening tool?
54 - skull shield
55 - shortbow
56 - longbow
57 - another bow, almost looks compound
58 - spiked warhammer
59 - sledgehammer or maul
5A - ruby-topped staff
5B - spear
5C - a staff with a blue globe in the crook
5D - a crescent moon signpost
5E - quilted armor
5F - rags
60 - um, quilted armor?
61 - spiked morningstar

62 - morningstar
63 - black tower shield
64 - thick brown and white bow
65 - wing-hilted sword
66 - leather armor
67 - plate armor
68 - grey robe
69 - blue and gold robe
6A - royal armor plate
6B - hieroglyphic inscribed stone
6C - wicked looking black axe
6D - gold half-circle axe
6E - odd axe
6F - headman's axe
70 - double-bladed axe
71 - red/gold lion shield
72 - blue eagle shield
73 - black and gold lion shield
74 - white, hooded cloak
75 - red cape
76 - nothing
77 - nothing

Random Seeds to Buy:

FF55h - leather armor of brilliance
0100h - mace of the moon, warhammer of structure, ring of accuracy
0300h - spiked club of slaying, +10 damage
0500h - valiant ring AC+19, ring of might, warhammer of the sky
0C00h - ring of health, other stuff
0F00h - fine longsword, 56% dam + 10%to hit, other stuff
1000h - vicious sword of the moon
1b00h - blue ring of light, leather armor of the mind

Godly Plate of the Ages

This ain't no plate to eat from. This is the information for the best(?) armor in the game. It was also sent to me by Meklar...

Here is how to change your armor (the on in the Armor Slot) to the best plate in the game (Godly Plate of the Ages AC75 AC Bonus 195% Indestructable). Note: This works on any item. See below for example.)

Offset:	Value:
B60	00
B61	00
B62	00
B63	01
B64	00
B67	4C
B68 to B6E	00
B6F	EC
B70	05
B71	05
B72	00
B73	3F
B74	01
B75	00
B76	00
B77	09
B78 to B7F	00

All of those previous addresses in previous mailings I've given you are useless unless the naming scheme is figured out. The true address is above the actual item data. It's hard to explain but I'll try.

For Example: (Notice this is for the armor slot. It is always the fourth line from the item name. And it begins at the second byte from the left on the fourth line. In this case it starts at B64. I've tried various numbers and have come up with items like Ruby Helm of the Heavens but have had no luck in figuring out what each address means individually.)

```
00000B60 0000 0001 0000 004C 0000 0000 0000 00EC .......L........
00000B70 0505 003F 0100 0009 0000 0000 0000 0000 ...?............
00000B80 0000 0001 0000 0000 0000 000D 0000 0001 ................
00000B90 0000 0060 0000 0010 0000 0000 0000 0000 ...`............
00000BA0 0000 0000 0000 0001 0000 0001 4675 6C6C ............Full
00000BB0 2050 6C61 7465 204D 6169 6C00 6F72 7461   Plate Mail.orta
00000BC0 6C00 0000 0000 0000 0000 0000 0000 0000 l...............
00000BD0 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000BE0 0000 0000 0000 0000 0000 0000 476F 646C ............Godl
00000BF0 7920 506C 6174 6520 6F66 2074 6865 2061 y Plate of the a
00000C00 6765 7300 2074 6865 2061 6765 7300 0000 ges. the ages...
00000C10 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000C20 0000 0000 0000 0000 0000 0000 0302 0097 ................
00000C30 0000 0064 1900 003C 9802 0000 0000 0000 ...d...<........
00000C40 0000 004B 0000 0000 0000 0000 0000 0000 ...K............
00000C50 0000 0000 0000 0000 0000 00FF 0000 00FF ................
00000C60 0000 0000 0000 0000 0000 00C6 0000 0000 ................
```

00000C70 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000C80 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000C90 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000CA0 0000 0054 0000 0000 0000 0000 0000 0000 ...T............
00000CB0 0000 0000 0000 0000 0000 0006 2500 0020 ....%..
00000CC0 1B00 0014 0000 0058 0200 0005 0000 005A ....X.......Z

In this way I have successfully turned every item in my inventory to Godly Plate of the Ages. Now the only probelm is finding out how to change it to Godly Plate of the Zodiac!

Ring Editing

This was sent by Thunder2:

Rings that give RESIST ALL MAX and +100 TO ALL ATTRIBUTES
Alterations to Enigma's *.DMP files
(Note:This is based on my character "Thunder2" Sorcerer)

In HEX using Hex Workshop:

For LEFT ring:
(Name:Godly ring of Zodiac-Optional)
000004B0 0000 0000 0000 0000 0000 0000 476F 646C
000004C0 7920 5269 6E67 206F 6620 7468 6520 7A6F
000004D0 6469 6163 0000 0000 0000 0000 0000 0000

(Change grapics of ring-Optional)
Alter last two Hex symbols (SEE Graphics Numbers Below)
000004F0 0000 0000 0000 0000 0000 0000 0503 000C

00000530 0000 0000 0000 0000 0000 0000 0000 0064
00000540 0000 0064 0000 0064 0000 0064 0000 0064
00000550 0000 0064 0000 0064 0000 0000 0000 0000
00000560 0000 0000 0000 0000 0000 0000 0000 0000
00000570 0100 0044 0000 0000 0000 0000 0000 0000
00000580 0000 0000 0000 0000 0000 000B 1B00 0040

For RIGHT ring:
(Name:Godly ring of Zodiac-Optional)
00000620 0000 0000 0000 0000 0000 0000 476F 646C
00000630 7920 5269 6E67 206F 6620 7468 6520 7A6F
00000640 6469 6163 0000 0000 0000 0000 0000 0000

(Change grapics of ring-Optional)
Alter last two Hex symbols (SEE Graphics Numbers Below)
00000660 0000 0000 0000 0000 0000 0000 0503 000C

000006A0 0000 0000 0000 0000 0000 0000 0000 0064
000006B0 0000 0064 0000 0064 0000 0064 0000 0064
000006C0 0000 0064 0000 0064 0000 0000 0000 0000
000006D0 0000 0000 0000 0000 0000 0000 0000 0000
000006E0 0100 0044 0000 0000 0000 0000 0000 0000
000006F0 0000 0000 0000 0000 0000 000B 1B00 0040

Attributes and magic resistance settings:
64 = Strength +100 for ring.

64 = Magic +100 for ring.
64 = Dexterity +100 for ring.
64 = Vitality +100 for ring.
64 = Fire Resistance MAX for ring.
64 = Lightning Resistance MAX for ring.
64 = Magic Resistance MAX for Ring.

* Hex number 40 gives it resist all*

(Ring Grapics HEX number)
07 Thick Ring
08 Ruby Ring
09 Thorny Ring
0A Sapphire Ring
0B Skull Ring
0C Metal Ring (Defalt)
0D Ebony Gold Ring
0E Orange Gem Gold Ring

A C Program To Decode Your *.DMP Files

The following program was sent to me by "raciper":

```
// Structure of Diablo character

#include <stdio.h>
#include <stdlib.h>

#define CORRECT_STRUCT_LENGTH 0x51e8

typedef struct
{
        long Id1;
        long Id2;
        long Id3;
        long Unknown1m; // 0
        long Unknown1n; // 0
        long Unknown1o; // 1
        long Unknown1p; // 0
        long Unknown1q; // 13
        long Unknown1r; // 1
        long Unknown1s; // 96
        long Unknown1t; // 16
        long Unknown1u; // 0
        long Unknown1v; // 0
        long Unknown1w; // 0
        long Unknown1x; // 1

        unsigned char Unknown2;
        char Description[64];
        char Name[64];
        unsigned char Unknown3[3];   //[0] = 1-7 : [1] 1-4
        long Unknown4a;
        long RawValue;
        long TrueValue;
        long MinDamage;
        long MaxDamage;
        long ArmorClass;
        unsigned char Spell;
        unsigned char Flags;
        unsigned char Byte1;
        unsigned char Byte2;
        long Unknown4b;
        long Unknown4c;
        long Unknown4d;
        long Unknown4e;
    long DurabilityLeft;
        long Durability;
        long Damage;
        long ToHit;
        long Armor;
        long Strength;
        long Magic;
        long Dexterity;
        long Vitality;
        long ResistFire;
        long ResistLightning;
        long ResistMagic;
        long Mana;
        long Life;
        long Unknown5x;
```

```
        long AdjDamage;
        long Radiance;
        long Unknown5a;
        long Unknown5b;
        long Unknown5c;
        long Unknown5d;
        long Unknown5e;
        long Unknown5f;
        long Unknown5g;
        long Unknown6a;
        long Unknown6b;
        long Unknown6c;
        long Unknown6d;
        long Unknown6e;
        char MinStr;
        char MinMag;
        char MinDex;
        char MinVit;
        long Unknown6g;
        long Unknown6h;
        long Unknown6i;
} EQUIPMENT;
// Unknown: Durability,
//      Knock Back,
//      Stop healing,
//      Invisablity
//      See invisable.
//      Rate of Fire, Range Adj, Homing Arrow, Fire Arrows, Lightning Arrows
//      Number Arrows per shot.
//      Modify Charges
//      Effect Spell Cost, Effect Spell Level
// Chaos? Loose all Mana, Exponetial Damage, Pox, Get hurt with use
// Damages attackers, Fear, Reduce trap damage
//

void DisplayEquipment(EQUIPMENT *e);

typedef struct _diablo_
{
  char            id[21];
  unsigned char SpellLevels[70];
  unsigned char SpellFlags[8];
  unsigned char Unknown1[48];
  char            HeroName[32];
  long            Level;
  long            Strength[2];
  long                    Magic[2];
  long                    Dexterity[2];
  long                    Vitality[2];
  long            LevelUp;
  long                    Unknown2a;
  long            Unknown2b;
  long            Life[4];
  long            Unknown2c;
  long            Mana[4];
  long            Unknown2d;
  long            Unknown2e;
  long            Experience;
  long            Unknown3[1];
  long            NextLevel;
  char            ResistUnknown;
```

```
  char              ResistMagic;
  char              ResistFire;
  char              ResistLightning;
  long              Gold;
  long                      Unknown4[107];
  EQUIPMENT         Head;
  EQUIPMENT         RightRing;
  EQUIPMENT         LeftRing;
  EQUIPMENT         Neck;
  EQUIPMENT         RightHand;
  EQUIPMENT         LeftHand;
  EQUIPMENT         Body;
  EQUIPMENT         Equipment[40];
  long              Unknown5;
  unsigned char Location[40];
  EQUIPMENT         Belt[8];
  char              eos[1];
} DIABLO;

DIABLO diablo;
int main(int argc, char *argv[])
{
          FILE *fp;
          long   len;
          int    count;
          if (sizeof(DIABLO)==CORRECT_STRUCT_LENGTH+36)
          {
                    printf("DIABLO--Length Correct of %d\n",sizeof(DIABLO));
          }
          else
          {
                    printf("DIABLO--Incorrect Length of %d (0x%04X), expecting %d (0x%04X)\n",
                              sizeof(DIABLO),sizeof(DIABLO),CORRECT_STRUCT_LENGTH+36,
CORRECT_STRUCT_LENGTH+36);
                    exit(0);
          }
          if (argc != 2)
          {
                    printf("Format:   ddump <dumpfile>\n");
                    exit(0);
          }
          fp = fopen(argv[1],"rb");
          if (!fp)
          {
                    printf("Error Opening: %s\n",argv[1]);
                    exit(0);
          }
          fseek(fp,0,SEEK_END);
          len = ftell(fp);\
          if (len != sizeof(diablo)-36)
          {
                    printf("\"%s\": invalid file length of %ld, expecting %d\n",argv[1],len,sizeof(diablo)-36);
                    fclose(fp);
                    exit(0);
          }
          fseek(fp,0,SEEK_SET);
          count = fread(&diablo,sizeof(diablo)-36,1,fp);
          fclose(fp);

          if (count==0)
          {
```

```
                printf("\"%s\": Read Error, too short\n",argv[1]);
        }
        for (count=0;count<sizeof(diablo.SpellLevels)/sizeof(char);count++)
        {
                if ((count&7)==0) printf("\nSpell Levels%3d:",count);
            printf("%3d(%02X)",diablo.SpellLevels[count],diablo.SpellLevels[count]);
        }
        printf("\n Spell Flags:");
        for (count=0;count<sizeof(diablo.SpellFlags)/sizeof(char);count++)
        {
                printf(" %02X",diablo.SpellFlags[count]);
        }
        printf("\n");
        for (count=0;count<sizeof(diablo.Unknown1)/sizeof(long);count++)
        {
            printf("Unknown1[%d]:%10d (%08X)\
n",count,diablo.Unknown1[count],diablo.Unknown1[count]);
        }
        printf("         Hero: \"%s\"",diablo.HeroName);
    printf("        Level: %ld\n",diablo.Level);
    printf("    Strength: %5ld/%-5ld\n",diablo.Strength[1], diablo.Strength[0]);
    printf("        Magic: %5ld/%-5ld\n",diablo.Magic[1],      diablo.Magic[0]);
    printf(" Dexterity: %5ld/%-5ld\n",diablo.Dexterity[1],diablo.Dexterity[0]);
    printf("    Vitality: %5ld/%-5ld\n",diablo.Vitality[1], diablo.Vitality[0]);
        printf("    Level Up: %5ld\n",         diablo.LevelUp);
        printf("Unknown 2a: %5ld (%08lX)\n",diablo.Unknown2a,diablo.Unknown2a);
        printf("Unknown 2b: %5ld (%08lX)\n",diablo.Unknown2b,diablo.Unknown2b);
        printf("         Life: %5ld/%-5ld (%ld/%ld)\n",
                (diablo.Life[0]+63)/64,(diablo.Life[1]+63)/64,
                (diablo.Life[2]+63)/64,(diablo.Life[3]+63)/64);
        printf("Unknown 2c: %5ld (%08lX)\n",diablo.Unknown2c,diablo.Unknown2c);
        printf("         Mana: %5ld/%-5ld (%ld/%ld)\n",
                (diablo.Mana[0]+63)/64,(diablo.Mana[1]+63)/64,
                (diablo.Mana[2]+63)/64,(diablo.Mana[3]+63)/64);
        printf("Unknown 2d: %5ld (%08lX)\n",diablo.Unknown2d,diablo.Unknown2d);
        printf("Unknown 2e: %5ld (%08lX)\n",diablo.Unknown2e,diablo.Unknown2e);
        printf("Experience: %5ld\n",diablo.Experience);
        for (count=0;count<sizeof(diablo.Unknown3)/sizeof(long);count++)
        {
            printf("Unknown3[%d]:%10ld\n",count,diablo.Unknown3[count]);
        }
        printf("Next Level: %5ld\n",diablo.NextLevel);
        printf("Resistances\n");
        printf("     Unknown: %3d%%\n",diablo.ResistUnknown);
        printf("        Magic: %3d%%\n",diablo.ResistMagic);
        printf("         Fire: %3d%%\n",diablo.ResistFire);
        printf(" Lightning: %3d%%\n",diablo.ResistLightning);
        printf("         Gold: %d\n",diablo.Gold);
        for (count=0;count<sizeof(diablo.Unknown4)/sizeof(long);count++)
        {
                printf("Unknown4[%d]:%10ld(%08lX)\
n",count,diablo.Unknown4[count],diablo.Unknown4[count]);
        }
        printf("\n");
        printf("           Head: ");DisplayEquipment(&diablo.Head);
        printf("       Ring 1: ");DisplayEquipment(&diablo.RightRing);
        printf("       Ring 2: ");DisplayEquipment(&diablo.LeftRing);
        printf("           Neck: ");DisplayEquipment(&diablo.Neck);
        printf(" Right Hand: ");DisplayEquipment(&diablo.RightHand);
        printf("    Left Hand: ");DisplayEquipment(&diablo.LeftHand);
        printf("           Body: ");DisplayEquipment(&diablo.Body);
```

```
	for (count=0;count<40;count++)
	{
		printf("Backpack%3d: ",count);
		DisplayEquipment(&diablo.Equipment[count]);
	}
	printf("    Unknown5:%10ld(%08lX)\n",diablo.Unknown5,diablo.Unknown5);
	for (count=0;count<sizeof(diablo.Location)/sizeof(char);count++)
	{
		if ((count%10)==0) printf("\n   Location:%3d:",count);
	    printf(" %02X",diablo.Location[count]);
	}
	printf("\n");
	for (count=0;count<8;count++)
	{
		printf("     Belt%3d: ",count);
		DisplayEquipment(&diablo.Belt[count]);
	}
	return 0;
}

void DisplayEquipment(EQUIPMENT *e)
{
	int i;
    printf("\n");
	if (e->Id1)		printf("          Id 1:%10ld (%08lX)\n",e->Id1,e->Id1);
	if (e->Id2)		printf("          Id 2:%10ld (%08lX)\n",e->Id2,e->Id2);
	if (e->Id3)		printf("          Id 3:%10ld (%08lX)\n",e->Id3,e->Id3);
	if (e->Unknown1m)	printf("    Unknown1m:%10ld (%08lX)\n",e->Unknown1m,e->Unknown1m);
	if (e->Unknown1n)	printf("    Unknown1n:%10ld (%08lX)\n",e->Unknown1n,e->Unknown1n);
	if (e->Unknown1o != 1)	printf("    Unknown1o:%10ld (%08lX)\n",e->Unknown1o,e->Unknown1o);
	if (e->Unknown1p)	printf("    Unknown1p:%10ld (%08lX)\n",e->Unknown1p,e->Unknown1p);
	if (e->Unknown1q !=13)	printf("    Unknown1q:%10ld (%08lX)\n",e->Unknown1q,e->Unknown1q);
	if (e->Unknown1r != 1)	printf("    Unknown1r:%10ld (%08lX)\n",e->Unknown1r,e->Unknown1r);
	if (e->Unknown1s !=96)	printf("    Unknown1s:%10ld (%08lX)\n",e->Unknown1s,e->Unknown1s);
	if (e->Unknown1t !=16)	printf("    Unknown1t:%10ld (%08lX)\n",e->Unknown1t,e->Unknown1t);
	if (e->Unknown1u)	printf("    Unknown1u:%10ld (%08lX)\n",e->Unknown1u,e->Unknown1u);
	if (e->Unknown1v)	printf("    Unknown1v:%10ld (%08lX)\n",e->Unknown1v,e->Unknown1v);
	if (e->Unknown1w)	printf("    Unknown1w:%10ld (%08lX)\n",e->Unknown1w,e->Unknown1w);
	if (e->Unknown1x != 1)	printf("    Unknown1x:%10ld (%08lX)\n",e->Unknown1x,e->Unknown1x);
    if (e->Unknown2)		printf("      Unknown2: %d (%02X)\n",e->Unknown2,e->Unknown2);
				printf("          Item: \"%s\" %s\n",e->Name,e->Description);
				printf("      Unknown3: %d %d %d\n",e->Unknown3[0],e->Unknown3[1],e->Unknown3[2]);
    if (e->Unknown4a)		printf("    Unknown4a:%10ld (%08lX)\n",e->Unknown4a,e->Unknown4a);
	if (e->RawValue || e->TrueValue)
					printf("         Value:%10ld/%-10ld\n",e->RawValue,e->TrueValue);
	if (e->MinDamage || e->MaxDamage)
					printf("        Damage:%10ld-%-10ld\n",e->MinDamage,e->MaxDamage);
```

```
        if (e->ArmorClass)        printf(" Armor Class:%10ld\n",e->ArmorClass);
        if (e->Spell)             printf("          Spell:%10d\n",e->Spell);
        if (e->Flags)             printf("          Flags: %02X\n",e->Flags);
        if (e->Byte1)             printf("         Byte 1: %3d (0x%02X)\n",e->Byte1,e->Byte1);
        if (e->Byte2)             printf("         Byte 2: %3d (0x%02X)\n",e->Byte2,e->Byte2);
        if (e->Unknown4b)         printf("    Unknown4b:%10ld (%08lX)\n",e->Unknown4b,e-
>Unknown4b);
        if (e->Unknown4c)         printf("    Unknown4c:%10ld (%08lX)\n",e->Unknown4c,e-
>Unknown4c);
        if (e->Unknown4d)         printf("    Unknown4d:%10ld (%08lX)\n",e->Unknown4d,e-
>Unknown4d);
        if (e->Unknown4e)         printf("    Unknown4e:%10ld (%08lX)\n",e->Unknown4e,e-
>Unknown4e);
        if (e->Durability)        printf("   Durability: %5ld/%-5ld\n",e->DurabilityLeft,e->Durability);
        if (e->Damage)            printf("         Damage: %+ld%%\n",e->Damage);
    if (e->ToHit)           printf("          ToHit: %+ld%%\n",e->ToHit);
        if (e->Armor)             printf("          Armor: %+ld%%\n",e->Armor);
        if ((e->Strength == e->Magic)&&(e->Dexterity==e->Vitality)&&(e->Magic==e->Vitality))
        {
                if (e->Magic)         printf(" All Attribs: %+ld\n",e->Magic);
        }
        else
        {
                if (e->Strength)      printf("      Strength: %+ld\n",e->Strength);
                if (e->Magic)         printf("         Magic: %+ld\n",e->Magic);
                if (e->Dexterity)     printf("     Dexterity: %+ld\n",e->Dexterity);
                if (e->Vitality)      printf("      Vitality: %+ld\n",e->Vitality);
        }
        if ((e->ResistFire == e->ResistMagic)&&(e->ResistFire==e->ResistLightning))
        {
                if (e->ResistFire)   printf(" Resists All: %+ld%%\n",e->ResistFire);
        }
        else
        {
                if (e->ResistFire)        printf(" Resist Fire: %+ld%%\n",e->ResistFire);
                if (e->ResistLightning) printf(" Resist Ltng: %+ld%%\n",e->ResistLightning);
                if (e->ResistMagic)       printf("Resist Magic: %+ld%%\n",e->ResistMagic);
        }
        if (e->Mana)              printf("           Mana: %+ld\n",e->Mana/64);
        if (e->Life)              printf("           Life: %+ld\n",e->Life/64);
        if (e->Unknown5x)         printf("    Unknown5x: %+ld (%08lX)\n",e->Unknown5x,e-
>Unknown5x);
        if (e->AdjDamage)         printf("   Adj Damage: %+ld\n",e->AdjDamage);
        if (e->Radiance)          printf("     Radiance: %+ld%%\n",e->Radiance*10);
        if (e->Unknown5a)         printf("    Unknown5a:%10ld (%08lX)\n",e->Unknown5a,e-
>Unknown5a);
        if (e->Unknown5b)         printf("    Unknown5b:%10ld (%08lX)\n",e->Unknown5b,e-
>Unknown5b);
        if (e->Unknown5c)         printf("    Unknown5c:%10ld (%08lX)\n",e->Unknown5c,e-
>Unknown5c);
        if (e->Unknown5d)         printf("    Unknown5d:%10ld (%08lX)\n",e->Unknown5d,e-
>Unknown5d);
        if (e->Unknown5e)         printf("    Unknown5e:%10ld (%08lX)\n",e->Unknown5e,e-
>Unknown5e);
        if (e->Unknown5f)         printf("    Unknown5f:%10ld (%08lX)\n",e->Unknown5f,e-
>Unknown5f);
        if (e->Unknown5g)         printf("    Unknown5g:%10ld (%08lX)\n",e->Unknown5g,e-
>Unknown5g);
        if (e->Unknown6a)         printf("    Unknown6a:%10ld (%08lX)\n",e->Unknown6a,e-
>Unknown6a);
        if (e->Unknown6b)         printf("    Unknown6b:%10ld (%08lX)\n",e->Unknown6b,e-
```

```
>Unknown6b);
        if (e->Unknown6c)       printf("    Unknown6c:%10ld (%08lX)\n",e->Unknown6c,e-
>Unknown6c);
        if (e->Unknown6d)       printf("    Unknown6d:%10ld (%08lX)\n",e->Unknown6d,e-
>Unknown6d);
        if (e->Unknown6e)       printf("    Unknown6e:%10ld (%08lX)\n",e->Unknown6e,e-
>Unknown6e);
        if (e->MinStr)          printf("Min Strength:%5d\n",e->MinStr);
        if (e->MinMag)          printf("    Min Magic:%5d\n",e->MinMag);
        if (e->MinDex)          printf("      Min Dex:%5d\n",e->MinDex);
        if (e->MinVit)          printf("Min Vitality:%5d\n",e->MinVit);
        if (e->Unknown6g)       printf("    Unknown6g:%10ld (%08lX)\n",e->Unknown6g,e-
>Unknown6g);
        if (e->Unknown6h)       printf("    Unknown6h:%10ld (%08lX)\n",e->Unknown6h,e-
>Unknown6h);
        if (e->Unknown6i)       printf("    Unknown6i:%10ld (%08lX)\n",e->Unknown6i,e-
>Unknown6i);
```