Diablo II Saved Game File Format

for Diablo II v1.09 and the Diablo II Expansion Set: Lord of Destruction

Updated April 2, 2004
visitors since

Introduction

I started out with just a page on the [item format](#). However, after starting a new [GUI-based saved game editor](#), and doing even more research on the "unknown" fields in the `.d2s` file, I discovered much more information on the rest of the file that isn't available anywhere else. So I'm publishing my findings.

File Header

(Note: all values larger than a byte are stored in x86 little-endian order -- i.e., least significant byte first. A "`short`" is 2 bytes long, and a "`long`" is 4 bytes long. Values starting with "0x" are given in hexadecimal notation; otherwise they are decimal.)

Byte Position	Size	Contents
0	`long`	File header. This must be the value 0xaa55aa55.

4	`long`	File version. The following values are known:
		   71 v1.00 through v1.06
		   87 v1.07 or Expansion Set v1.08
		   89 standard game v1.08
		   92 v1.09 (both the standard game and the Expansion Set.)
		This document only covers version 92 of the file format.

8	`long`	File size.
12	`long`	Checksum. If you attempt to hack the file without storing the correct checksum afterwards, *your game will fail to load!* Fortunately, the checksum algorithm is a relatively simple one. After clearing the checksum field, you add up the values of all the bytes in the file, rotating the running total one bit to the left before adding each byte. Then store the result in this field. *Thanks go to the [Shadowmaster](#) for publishing the checksum code.*
16	`long`	*Unknown; the value can be either 0 or 1*
20	16 `chars`	[Character name](#). The name may be up to 15 characters long; the rest of the field must be padded with null bytes. *Remember the rules for Diablo II character names: 2-15 characters, containing only upper and lower case letters (A-Z), with the possible addition of one dash (-) or underscore (_) as long as it is not the first or last character of the name.*
		Character status. This is a bit field:

7	6	5	4	3	2	1	0
unknown		Expansion Character	*unknown*	Died	Hardcore	*unknown*	

36	byte	*Note: the "died" bit apparently indicates that your character has died at some point in the past. It is never cleared when you resume the game. To find out whether your character is currently dead, I think you need to look in the item list below to see if there is some corpse data.* *Thanks go to Guillaume Courtin of France for pointing out the Hardcore bit while I was busy with other things.*

37	byte	Character progression. This number tells (sort of) how many acts you have completed from all difficulty levels. It appears to be incremented when you kill the final demon in an act -- i.e., Andarial, Duriel, Mephisto, and Diablo / Baal. There's a catch to that last one: in an Expansion game, the value is *not* incremented after killing Diablo, but is incremented by 2 after killing Baal. (The reason is unknown.) So it skips the values 4, 9, and 14. I believe this value is used in determining your character's title. The title is one of the following values (depending on the character class' gender):

Value	Standard	Hardcore	Value	Expansion	Hardcode Exp.
0-3	*(no title)*		0-3	*(no title)*	
4-7	Sir / Dame	Count / Countess	5-8	Slayer	Destroyer
8-11	Lord / Lady	Duke / Duchess	10-13	Champion	Conqueror
12	Baron / Baroness	King / Queen	15	Patriarch / Matriarch	Guardian

38	2 bytes	*unknown*

40	byte	Character class. The defined classes are:
		  0 Amazon
		  1 Sorceress
		  2 Necromancer
		  3 Paladin
		  4 Barbarian
		  5 Druid (Expansion character only)
		  6 Assassin (Expansion character only)

41	2 bytes	*unknown; I've only seen the values { 16, 30 } here.*
43	byte	Character's level. This is the level shown on the character selection screen, but it should equal the level given in the character statistics section.
44	long	*unknown*
48	long	Time stamp. This is in the standard `time()` format of the number of seconds which have elapsed since midnight, January 1, 1970 (UTC).
52	long	*unknown*
		Thanks go to Ronnie Brohn for pointing out this section. These are the skill ID's assigned to the hotkeys for Skill 1 through Skill 16. (Note that Skills 8 and above is only available in the Expansion Set.) If a skill hotkey is

56	16 longs	not assigned to a skill, the value is `0xffff`. Hotkey definitions are stored in the `character.key` file. The structure of that file is not covered by this document (and I see no need to modify it outside of the game.)
120	long	The action assigned to the left mouse button. The value of this field is a <u>skill ID</u>.
124	long	The action assigned to the right mouse button.
128	long	In an Expansion character, the action assigned to the alternate left mouse button. (The button assignments are swapped when you swap weapons.)
132	long	In an Expansion character, the action assigned to the alternate right mouse button.
136	32 bytes	*unknown*
168	3 bytes	These bytes indicate which difficulty the character is playing. The first byte corresponds to Normal, the second Nightmare, and the third Hell. If the value is zero, the character is not playing at that level. Otherwise, the value looks like this: | 7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---
Active	*unknown*				Which Act the character is in (0-4)			
171	long	Map ID. This value looks like a random number, but it corresponds with one of the `longwords` found in the `character.map` file, according to the difficulty being played.
175	short	*unknown*
177	short	*uncertain;* it looks like this is set on an Expansion character whose mercenary has died.
179	4 bytes	This looks like a random ID for your mercenary. It is 0 if you have never had a mercenary. If your mercenary has died or (in the standard game) been left behind when you move on to the next act, this field is still set to the last mercenary you had.
183	2 bytes	This is a numerical index into the game's language-dependent string table for mercenary names. There is a separate list for each type of mercenary (Rogue Scout, Desert Mercenary, Eastern Sorceror, and Barbarian). I've written up a <u>list of mercenary names</u> for the English 1.09 patch.
185	short	This code determines the difficulty level and act where your mercenary is found, as well as the attribute of your mercenary (i.e., Cold, Fire, Lightning). I have a <u>list of the mercenary codes</u> appended to the end of the mercenary name list.
187	long	Your mercenary's experience points.
191	144 bytes	*unknown*

Quest Completion Data

The quest data begins with the following header:

Byte Position	Size	Contents

| 335 | 4 chars | The string identifier "woo!". *(I guess somebody at Blizzard had a sense of humor. :)* |
| 339 | 6 bytes | *unknown; I always see the bytes { 6, 0, 0, 0, 42, 1 } here.* |

The header is followed by three structures, one for each difficulty level. *Remember, the byte offsets given here are offsets into the structure; the first structure is at offset 345 in the file.*

Although there is some variation in the meaning of the bits per quest, some of the bits appear to have constant meaning.

Bit 0 indicates the quest is complete. If bit 0 is clear but the rest of the field is non-zero, then the quest has been started, but not finished.

Bit 1 generally means you have completed the requirements for the quest (i.e., killed the boss demon), and all that's left is to collect the reward -- for example, "Charsi will imbue an item with magical power." Not all quests have this option. If this bit is set, bit 0 must be clear.

Bit 2 is often set when an NPC gives you a quest.

Bit 12 is set when you have seen the swirling fire animation that closes a quest icon.

Bit 13 indicates the quest was completed in the current game; when you save the game and then reload it, all bit 13's are cleared.

Byte Position	Size	Contents		
0	short	This field contains a 1 if you have been introduced (by Warriv) to Act I.		
2	6 shorts	These fields contain quest completion data for each quest in Act I.		
		short #	**Quest**	**Notes**
		0	Den of Evil	Bit 4 is set when you enter the Den.
		1	Sisters' Burial Grounds	Bit 4 is set when you enter the Burial Grounds.
		2	Tools of the Trade	Bit 6 is set when you pick up the Horadric Malus.
		3	The Search for Cain	Bit 4 is set when you enter Tristram. Bit 10 indicates whether you have completed the secret Cow Level ("Moo"). If you want to fight the Cow King again, just clear this bit! If you enter Act II without rescuing Deckard Cain, bit 14 will get set. You will not be able to rescue Cain yourself; the Rogues will have done it instead, and as a consequence, you will be charged a fee if you want Cain to identify items for you.
		4	The Forgotten Tower	Bit 2 is set when you read the Moldy Tome. Bit 6 is set when you enter the Forgotten Tower.

		5	Sisters to the Slaughter	

14	short	*uncertain; I think this gets set to a non-zero value after you travel from Act I to Act II.*		
16	short	This field contains a 1 if you have been introduced (by Jerhyn) to Act II.		

18	6 shorts	These fields contain quest completion data for each quest in Act II.		

short #	Quest	Notes
0	Radament's Lair	Bit 4 is set when you find Radament.
1	The Horadric Staff	Bit 4 is set when Cain tells you about the Viper Amulet. Bit 5 is set when Cain tells you about the Staff of Kings. Bit 10 is set when Cain tells you about the Horadric Staff. Bit 11 is set when you make the Horadric Staff.
2	Tainted Sun	Bit 2 is set when the sun goes out. Bit 3 is set when Drognan tells you why.
3	Arcane Sanctuary	
4	The Summoner	
5	The Seven Tombs	Bit 3 is set when you talk to Tyrael. Bit 4 is set when you talk to Jerhyn (after killing Duriel). Bit 5 is set when you kill Duriel. Bit 6 is set when Atma congratulates you. Bit 7 is set when Warriv `''`. Bit 8 is set by Drognan. Bit 9 is set by Lysander. Bit 10 is set by Cain. Bit 11 is set by Fara.

30	short	*uncertain; I think this gets set to a non-zero value after you travel from Act II to Act III.*		
32	short	This field contains a 1 if you have been introduced (by Hratli) to Act III.		

34	6 shorts	These fields contain quest completion data for each quest in Act III.		

short #	Quest	Notes
0	Lam Esen's Tome	
1	Khalim's Will	
2	Blade of the Old Religion	Bit 2 is set when you pick up the Gidbinn. Bit 3 is set when Hratli asks you to find the Gidbinn.
3	The Golden Bird	Bit 2 is set when Cain tells you about the Jade Figurine. Bit 4 is set when Cain tells you about the Golden Bird. Bit 5 is set when you are given the Potion of Life, and cleared again when you drink the Potion. (This prevents you from drinking

		more than one in a game.) Bit 6 is set when you find the Jade Figurine.
	4	The Blackened Temple
	5	The Guardian

46	short	*uncertain; I think this gets set to a non-zero value after you travel from Act III to Act IV.*
48	short	This field contains a 1 if you have been introduced to Act IV.

50	3 shorts	These fields contain quest completion data for each quest in Act IV. Note that there are only three quests here, as opposed to 6 for the first three Acts.

short #	Quest	Notes
0	The Fallen Angel	
1	Terror's End	
2	Hell's Forge	

56	short	*uncertain; I think this gets set to a non-zero value after you travel from Act IV to Act V in an Expansion game,* **but I have not yet verified this assumption.** *It appears to be zero after completing a standard game.*
58	3 shorts	*unknown; maybe act IV has 6 fields for quests after all, even though only 3 are used?*
64	short	*unknown; in an Expansion character, this was set to 1 after completing Terror's End and talking to Cain in act IV*
66	2 shorts	*unknown*

70	6 shorts	These fields contain quest completion data for each quest in Act V.

short #	Quest	Notes
0	Siege on Harrogath	Bit 3 is set when you find Shenk. Bit 5 is set when Larzuk offers to socket an item for you.
1	Rescue on Mount Arreat	
2	Prison of Ice	Bit 7 is set when you read the Scroll of Resistance. Bit 8 is set after you rescue Anya and talk to Malah.
3	Betrayal of Harrogath	Bit 4 is set when Anya offers to personalize an item for you.
4	Rite of Passage	
5	Eve of Destruction	Bit 4 is set when Larzuk congratulates you. Bit 5 is set when Cain congratulates you. Bit 6 is set when Malah `''`. Bit 7 is set by Tyrael. Bit 8 is set by Qual-Kehk. Bit 9 is set by Anya.

82	7 shorts	*unknown*

Waypoint Data

The waypoint data begins with the following header:

Byte Position	Size	Contents
633	2 `chars`	The string identifier "`ws`".
635	6 bytes	*unknown*

The header is followed by three structures, one for each difficulty level. *Remember, the byte offsets given here are offsets* into the structure; *the first structure is at offset 641 in the file*.

Byte Position	Size	Contents
0	2 bytes	*unknown; I always see the values { 2, 1 } here.*
2	5 bytes	Waypoints. This is a bitfield, with one bit assigned to each waypoint in LSB order -- so bit 0 in the Rogue Encampment waypoint for Act I. The first waypoint in every Act is activated as soon as you enter that Act. There are 9 waypoints (bits) in each of Acts I, II, and III, and 3 waypoints (bits) in Act IV, so the last waypoint before Diablo (River of Flame) is bit 29 (since we start counting from 0). The first waypoint for Act V follows at bit 30, and continues to the last (ninth) waypoint in Act V at bit 38.
7	17 bytes	*unknown*

NPC Introductions

My data on the next section is very sketchy... help?

Byte Position	Size	Contents
714	2 `chars`	The string identifier "`w4`".
716	1 byte	*unknown*
717	1 byte	You have been introduced to: | 7	6	5	4	3	2	1	0 | | Warriv | | Charsi | | Kashya | Akara | Gheed | |
718	1 byte		7	6	5	4	3	2	1	0 | | Greiz | | Meshif | Geglash | Lysander | Fara | Drognan | |
719	1 byte		7	6	5	4	3	2	1	0 | | Alkor | | Asheara | | | Cain | | Elzix |
720	1 byte		7	6	5	4	3	2	1	0 | | Malah | Anya | | Natalya | Meshif | | | Ormus |
721	1 byte		7	6	5	4	3	2	1	0 | | | | | | | Cain | Qual-Kehk | Nihlathak |

722	3 bytes	*unknown*
725	8 bytes	Introductions repeated for Nightmare difficulty.
733	8 bytes	Introductions repeated for Hell difficulty.

741	1 byte	It would appear that bits 1-6 of byte 741 get set after you take the caravan to Act II in Normal difficulty. Bit 7 of byte 741 through bit 1 of byte 743 get set after you sail to Act III. On entering Act IV, bits 2, 5, 6, & 7 of byte 743 and bits 0, 3, & 4 of byte 744 get set. When you return to a previous act and talk to the NPC's, these bits are cleared. You have yet to be welcomed back by:

7	6	5	4	3	2	1	0
Warriv	?	Charsi	Warriv	Kashya	Akara	Gheed	

742	1 byte	

7	6	5	4	3	2	1	0
Greiz	Jerhyn	Meshif	Geglash	?	Fara	Drognan	?

743	1 byte	

7	6	5	4	3	2	1	0
Alkor	Hratli	Asheara			?	?	Elzix

744	1 byte	

7	6	5	4	3	2	1	0
			?	?			Ormus

745	4 bytes	*unknown*
749	8 bytes	Greetings repeated for Nightmare difficulty.
757	8 bytes	Greetings repeated for Hell difficulty.

Character Statistics

The character statistics begin with the following header:

Byte Position	Size	Contents
765	2 `chars`	The string identifier "`gf`".
767	`short`	This bit field indicates the presence or absence of each particular statistic. If the bit is 0, the corresponding statistic is zero and not stored in the file. If the bit is 1, the statistic has a `long` value stored in the file. Note that many of the statistics are never zero, so they will always be present; but there is a bit assigned to them nonetheless. The bits are assigned as follows: {table below}

Bit	Statistic
0	Strength
1	Energy
2	Dexterity
3	Vitality
4	Stat Points Remaining
5	Skill Choices Remaining

6	Life (current)
7	Life (base)
8	Mana (current)
9	Mana (base)
10	Stamina (current)
11	Stamina (base)
12	Level
13	Experience
14	Gold in Inventory
15	Gold in Stash

From this point on, the contents of the file are (more or less) variable, because the contents of the statistics section are optional.

Field	Details
Strength	These fields are self-explanatory. They are always present. Remember, all values are stored as `longs`.
Energy	
Dexterity	
Vitality	
Stat Points Remaining *(optional)*	The number of Stat Points earned but not distributed.
Skill Choices Remaining *(optional)*	The number of Skill Choices earned but not distributed.
Life (current)	These fields are (usually) always present. (There is an exception: if your character is dead, the current Life field will be gone!) They are also *not* plain integer values! Instead, each field is a fixed-point binary number, with a 24-bit integer part and an 8-bit fraction part. For example, if the life (current) field contained the value 0x020ac0, then to get the amount of life remaining you would take that value and divide by 256.0 to get 522.75 (rounding the number to an integer for display ... although I'm not certain in which direction the number is rounded). Note that the current amount of life (/mana /stamina) may be more than the base amount, because the base does not take into account any blessings bestowed by magical items you are carrying.
Life (base)	
Mana (current)	
Mana (base)	
Stamina (current)	
Stamina (base)	
Level	Your character's level. This value must be in the range 1-99 (and is therefore always present, even on a new character) and should be the same as byte 43 in the file header.
Experience *(missing on a new character)*	The amount of experience your character has. If you haven't killed a single monster in the game, your experience will be 0, and this field is not stored. Otherwise, this field is always present.

Gold in Inventory *(optional)*	The amount of gold you are carrying. Just as a helpful reminder, the maximum amount of gold you may carry is directly proportional to your level, at 10,000 gold per level. Thus, a new character can only carry 10,000 gold pieces (which is still a lot, if you think about it, even if each piece is smaller than a dime *:)*, but a level 99 character (the limit) can carry nearly a million in gold (990,000).
Gold in Stash *(optional)*	The amount of gold you have stowed away. Just as a helpful reminder, the maximum amount of gold you may store in the stash is related to your level as follows:

Level	Max. Gold
1-30	$\texttt{int}(level \div 10 + 1) \times 50{,}000$
31-99	$\texttt{int}(level \div 2 + 1) \times 50{,}000$

Character Skills

The character skills section begins with the 2-`character` header, "`if`". This is followed by 30 bytes, each byte corresponding to one of the character's special skills. To save space, I'll list the skills in a separate table.

Item List

The next major section of the `.d2s` file is the item list. It begins with the following header:

Size	Contents
2 `chars`	The string identifier "`JM`".
`short`	The number of items your character has. This includes items equipped, tucked in your belt, stored in your inventory, stored in your stash, and hidden in the Horadric Cube. It does *not*, however, include gems, runes, or jewels which have been inserted into a socketed item. (Those are counted as part of the item.)

This header is followed by a list of items. The format of the items is **way** beyond the scope of this document; refer to my Item Format page for a detailed description.

After the list of items, you will find another item list header similar to the one shown above; only this time the item count will be 0. Usually. *It looks like if your character has died, then the item count in this second header will be 1, and 12 bytes of (unknown) data will follow, after which there will be another header and item list for items on your corpse. Sadly, I haven't died often enough to test this theory, much less figure out what those extra bytes mean.*

If you have an Expansion character, then the corpse item list will be followed by the 2-`character` header "`jf`". If *and only if* you have a mercenary (alive or dead), this header is followed by an item list header and (possibly empty) item list containing items equipped on the mercenary. This item list is followed by the trailer `characters` "`kf`".

If you have a necromancer, it is possible for you to have an Iron Golem that is preserved when your game is saved and restored. The Iron Golem is based on an item. Following the mercenary item list, there will be a single byte that is 0 if there is no golem, or 1 if there is. If there is a golem, this byte is followed by a single item. Note that this appears to be available only on Expansion characters.

The End

Please send any additions or corrections to me using my [web-based mail form](#).