BULBAPEDIA

Pokémon data structure (Generation III)

From Bulbapedia, the community-driven Pokémon encyclopedia.

Pokémon in the Pokémon Ruby and Sapphire, FireRed and LeafGreen, and Emerald Versions are all stored the same way in a 100-byte structure. All numbers are stored in little-endian order.

Contents

Notes

Personality value

The personality value controls many things, including gender, Unown's letter, Spinda's dots, any Pokémon's Nature, and more.

Pokémon				
	type	offset	length (in bytes)	offset (decimal)
Personality value	u32	0x00	4	0
OT ID	u32	0x04	4	4

Nickname	u8[10]	0x08	10	8
Language	u8	0x12	1	18
Misc. Flags	u8	0x13	1	19
OT name	u8[7]	0x14	7	20
Markings	u8	0x1B	1	27
Checksum	u16	0x1C	2	28
????	u16	0x1E	2	30
Data	u8[48]	0x20	48	32
Status condition	u32	0x50	4	80
Level	u8	0x54	1	84
Mail ID	u8	0x55	1	85
Current HP	u16	0x56	2	86
Total HP	u16	0x58	2	88
Attack	u16	0x5A	2	90
Defense	u16	0x5C	2	92
Speed	u16	0x5E	2	94
Sp. Attack	u16	0x60	2	96
Sp. Defense	u16	0x62	2	98

OT ID

The Original Trainer's ID number. This number is part of the XOR encryption key for the data section, and is also used in Shiny determination and the lottery. The first 2 bytes of this number are the Trainer ID visible on the status screen. The final 2 bytes are the Secret ID of the trainer that caught it.

Nickname

The Pokémon's nickname, limited to 10 characters. The characters represented by each byte are determined by the proprietary character set.

Language

The language of origin is language of the game the Pokémon originates from.

In Western languages, a Pokémon's language of origin determines which font to use to display its name and Original Trainer. This allows the names and Original Trainers of Pokémon from Japanese games to display correctly, including displaying Latin letters as fullwidth characters.

In Japanese, the language of origin is entirely ignored—names are always rendered using the Japanese character set. This causes all names to be truncated to five characters (even though they can be up to 10 characters in Western languages). In some cases, this causes characters to render as mojibake; for example, if the in-game trade Seel from Spanish Pokémon FireRed and LeafGreen (whose nickname is normally SEELÍN) is traded to a Japanese game, its nickname will be displayed as ＳＥＥＬ⊐.

The values that the languages correspond to are:

Index	Language
1	Japanese
2	English
3	French
4	Italian
5	German
6	*unused*
7	Spanish

In the Generation III games, Eggs always have their language set to Japanese. This does not cause any issues as, upon hatching, the language of the Egg is set to the language of the game it hatched in.

Misc. Flags

This byte houses 4 flags:

- **Is Bad Egg (Bit 0):** When this flag is set, the Pokémon will be treated as a Bad Egg. If a Pokémon's checksum is invalid, this flag is set, marking it as a Bad Egg and making it unusable.
- **Has Species (Bit 1):** This flag is set whenever the Pokémon species index is non-zero, which should be the case for any Pokémon. It is used as a sanity check for empty spaces, and any Pokémon without this flag set cannot be bred and will disappear when group selected.
- **Use Egg Name (Bit 2):** When this flag is set, the Pokémon will ignore their nickname and display the game's regional variant of "EGG". Only eggs should have this flag set. Note that this flag is independent from the egg flag in the subdata structure.
- **Block Box RS (Bit 3):** When this flag is set, the Pokémon cannot be deposited into Pokémon Box Ruby & Sapphire. This flag likely had a broader purpose but, in practice, this is its only known effect.
- **Bits 4-7**: These bits are unused, and are just padding for the other flags. They should be set to 0.

OT name

The name of the Pokémon's Original Trainer. The characters represented by each byte are determined by the proprietary character set.

Markings

The markings seen in the storage Box. These markings serve only to aid in organizing large collections of Pokémon.

Bit	Mark
0	●
1	■
2	▲
3	♥

Checksum

The checksum for the 48-byte data section of this structure. It is computed by adding all of the unencrypted values of that section one word at a time. If the computed sum and the stored checksum do not match, the Pokémon is interpreted as a Bad Egg.

????

Unknown, possibly simply padding (not used and usually set to either 0 or -1, depending on the data type).

Data

Certain data pertaining to the Pokémon that is stored in a special and encrypted format.

Status condition

The Pokémon's status condition is stored as follows:

Bit		Status
0-2	SLP	Sleep
3	PSN	Poison
4	BRN	Burn
5	FRZ	Freeze
6	PAR	Paralysis
7	PSN	Bad Poison

The three sleep bits are used to indicate turns of sleep. So 111_2 = 7 turns of sleep, 101_2 = 5 turns, et cetera.

Mail ID

In the Gen 3 games, when a player receives a Pokémon holding a mail item (whether from an in-game trade or from a friend) the message tied to that mail is not stored not on the Pokémon itself, but instead somewhere else in the trainer's save data. Each of these messages has a unique ID associated with it. This byte stores that ID. If the Pokémon is not holding a mail item, this byte is set to 0xFF (255).

Data location

This section is incomplete.
Please feel free to edit this section to add missing information and complete it.
Reason: Are the addresses below only for US games? Also, is the mentioned "general region" of box data correct?

A Trainer's party starts at the following addresses in the GBA's RAM.

Game	Address
Ruby Sapphire	0x03004360
Emerald	0x02024190 0x020244EC[US,FR]
FireRed	0x02024284

LeafGreen	0x020241E4 0x02024284[US]

An opponent's party, or a wild Pokémon, starts at the following addresses.

Game	Address
Ruby Sapphire	0x030045C0
Emerald	0x02024744
FireRed LeafGreen	0x0202402C

The 600 bytes following these addresses describe a whole team of 6 Pokémon.

The full 100-byte structure for a Pokémon is only used to describe Pokémon being held in the player's party. When Pokémon are stored in the PC, their data is recorded using only the first 80 bytes of this structure, stopping after the data field. The last 20 bytes (excepting status condition, current HP, and Pokerus remaining byte) can all be recalculated from data in the data substructure when a Pokémon is withdrawn (level being derived from experience). This also explains why Pokémon suffering a status condition are "cured" when put in the PC.

This means there are also 33,600 bytes (80 bytes * 30 per Box * 14 Boxes) elsewhere in the GBA's RAM describing Pokémon in the PC. When the GBA's saved state (including memory contents) is unzipped into a 740,000+ byte file and viewed, the 14 Boxes of 420 Pokémon are stored in the general region of $038000 and $040000. In the US version of Pokémon Emerald, box data is between 0x02FE9888 and 0x02FF1BC8, non-inclusive. (Some emulators may address their RAM slightly differently.) The first 6 80-byte structures make up, from left to right, the first row of Pokémon in box 1. The next Pokémon gets placed on the next row. After 5 rows (30 80-byte structures), the next Pokémon is placed in box 2, and so on.

See also

- Pokémon data substructures (Generation III)

Links

- PokemonMakerV4x Help and 80 bytes Make a Pokémon (http://www.ppnstudio.com/maker/PokemonMakerHelp.txt)
- pokemon.h | pokeemerald decomp (https://github.com/pret/pokeemerald/blob/master/include/pokemon.h)

Related articles

Data structure in the Pokémon games

General	Character encoding
Generation I	Pokémon species • Pokémon • Poké Mart • Character encoding (Stadium) • Save
Generation II	Pokémon species • Pokémon • Trainer • Character encoding (Stadium • Korean) • Save
Generation III	Pokémon species (Evolution • Pokédex • Type chart) Pokémon (substructures) • Move • Contest • Contest move • Item Trainer Tower • Battle Frontier • Character encoding (GameCube) • Save
Generation IV	Pokémon species (Evolution • Learnsets) Pokémon • Save • Character encoding (Wii)
Generation V–present	Character encoding
Generation VIII	Save
TCG GB and GB2	Character encoding

This data structure article is part of **Project Games**, a Bulbapedia project that aims to write comprehensive articles on the Pokémon games.

Retrieved from "https://bulbapedia.bulbagarden.net/w/index.php?title=Pokémon_data_structure_(Generation_III)&oldid=4433487"