Turbo ParTool - Krótka instrukcja

By Atlas

8 luty 2026

1. Export pliku .par za pomocą PARex

PARex jest to program służący do eksportu pliku `.par` z **KnightShift** do postaci katalogu z plikami `.cpp`, które reprezentują każdą sekcję obiektów pliku `.par`.

Jak używać?:

1. Włączamy program.
2. Wpisujemy nazwę pliku wejściowego wraz z formatem.
3. Klikamy przycisk enter i czekamy.
4. Po wyłączeniu okna konsoli wiemy, że program skończył pracę.

Program działa również w trybie ARGC&ARGV. Przykład użycia:

```
PARex.exe <nazwa_pliku.par>
```

Szybki Export:

Aby w szybki sposób dekompilować plik `.par` należy korzystać z `EXPORT.ps1` lub `EXPORT.bat`. W celu modyfikacji argumentów/konfiguracji należy zedytować plik `.bat/.ps1` przez notatnik.

Przkład pliku .bat:

```
PARex.exe <nazwapliku.par>
```

Przykład pliku .ps1:

```
.\PARex.exe .\<nazwapliku.par>
```

2. Opcjonalna konfiguracja PARex

Plik **PARex.cfg** umożliwia konfigurację sposobu eksportu danych do plików `.cpp`. Użytkownik może zdecydować, czy wybrane wartości będą prezentowane jako pojedyncze bity (flagi) o określonych etykietach, czy w formacie surowym. W przypadku wyłączenia mapowania na flagi, dane zostaną zapisane jako typ `uint32_t`. Rezygnacja z flagowania pozwala na znaczące przyspieszenie procesu eksportu oraz skrócenie czasu kompilacji wygenerowanego kodu. W pliku konfiguracyjnym dostępne są następujące parametry:

Prawda:

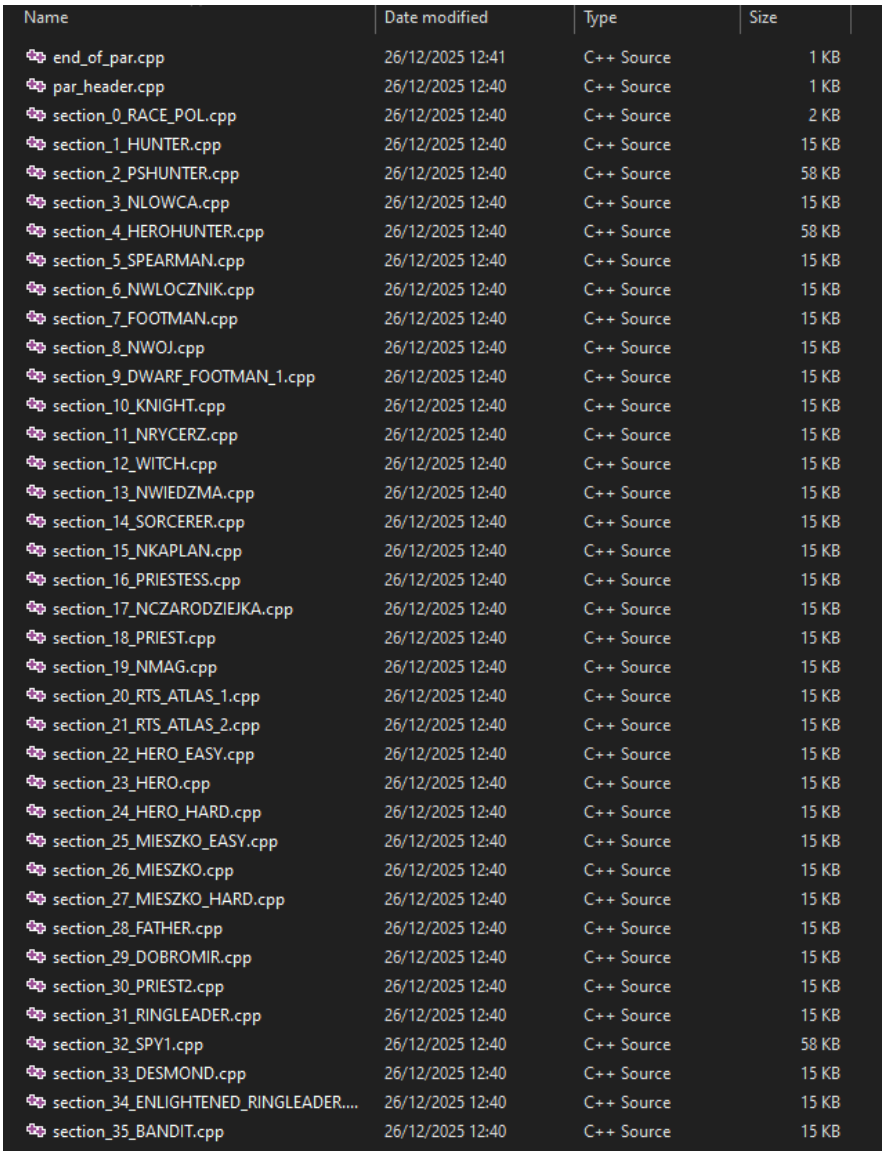
- YES/Yes/yes
- TRUE/True/true
- 1

Fałsz:

- NO/No/no
- FALSE/False/false
- 0

3. Ogólny pogląd na strukturę otrzymanych plików po eksporcie/dekompilacji

Po eksporcie pliku `.par`, otrzymamy katalog w którym znajdują się pliki `.cpp` reprezentujące sekcje z obiektami. Składnia plików jest w języku `C++`. Oprócz plików `.cpp` z sekcjami - otrzymujemy również plik `section_order.txt`, w którym znajduje się kolejność kompilacji plików sekcji.



Name	Date modified	Type	Size
end_of_par.cpp	26/12/2025 12:41	C++ Source	1 KB
par_header.cpp	26/12/2025 12:40	C++ Source	1 KB
section_0_RACE_POL.cpp	26/12/2025 12:40	C++ Source	2 KB
section_1_HUNTER.cpp	26/12/2025 12:40	C++ Source	15 KB
section_2_PSHUNTER.cpp	26/12/2025 12:40	C++ Source	58 KB
section_3_NLOWCA.cpp	26/12/2025 12:40	C++ Source	15 KB
section_4_HEROHUNTER.cpp	26/12/2025 12:40	C++ Source	58 KB
section_5_SPEARMAN.cpp	26/12/2025 12:40	C++ Source	15 KB
section_6_NWLOCZNIK.cpp	26/12/2025 12:40	C++ Source	15 KB
section_7_FOOTMAN.cpp	26/12/2025 12:40	C++ Source	15 KB
section_8_NWOJ.cpp	26/12/2025 12:40	C++ Source	15 KB
section_9_DWARF_FOOTMAN_1.cpp	26/12/2025 12:40	C++ Source	15 KB
section_10_KNIGHT.cpp	26/12/2025 12:40	C++ Source	15 KB
section_11_NRYCERZ.cpp	26/12/2025 12:40	C++ Source	15 KB
section_12_WITCH.cpp	26/12/2025 12:40	C++ Source	15 KB
section_13_NWIEDZMA.cpp	26/12/2025 12:40	C++ Source	15 KB
section_14_SORCERER.cpp	26/12/2025 12:40	C++ Source	15 KB
section_15_NKAPLAN.cpp	26/12/2025 12:40	C++ Source	15 KB
section_16_PRIESTESS.cpp	26/12/2025 12:40	C++ Source	15 KB
section_17_NCZARODZIEJKA.cpp	26/12/2025 12:40	C++ Source	15 KB
section_18_PRIEST.cpp	26/12/2025 12:40	C++ Source	15 KB
section_19_NMAG.cpp	26/12/2025 12:40	C++ Source	15 KB
section_20_RTS_ATLAS_1.cpp	26/12/2025 12:40	C++ Source	15 KB
section_21_RTS_ATLAS_2.cpp	26/12/2025 12:40	C++ Source	15 KB
section_22_HERO_EASY.cpp	26/12/2025 12:40	C++ Source	15 KB
section_23_HERO.cpp	26/12/2025 12:40	C++ Source	15 KB
section_24_HERO_HARD.cpp	26/12/2025 12:40	C++ Source	15 KB
section_25_MIESZKO_EASY.cpp	26/12/2025 12:40	C++ Source	15 KB
section_26_MIESZKO.cpp	26/12/2025 12:40	C++ Source	15 KB
section_27_MIESZKO_HARD.cpp	26/12/2025 12:40	C++ Source	15 KB
section_28_FATHER.cpp	26/12/2025 12:40	C++ Source	15 KB
section_29_DOBROMIR.cpp	26/12/2025 12:40	C++ Source	15 KB
section_30_PRIEST2.cpp	26/12/2025 12:40	C++ Source	15 KB
section_31_RINGLEADER.cpp	26/12/2025 12:40	C++ Source	15 KB
section_32_SPY1.cpp	26/12/2025 12:40	C++ Source	58 KB
section_33_DESMOND.cpp	26/12/2025 12:40	C++ Source	15 KB
section_34_ENLIGHTENED_RINGLEADER....	26/12/2025 12:40	C++ Source	15 KB
section_35_BANDIT.cpp	26/12/2025 12:40	C++ Source	15 KB

Rysunek 1: Lista plików .cpp po eksporcie.

4. Edycja plików:

- Pliki `.cpp` dobrze jest edytować w programie geany, który w estetyczny sposób przedstawia sekcję w postaci drzewka.
- Plik `section_order.txt` to plik, w którym jest kolejność odpowiednich sekcji. Przy dodawaniu nowej sekcji należy wpisać nowo dodaną sekcję do tego pliku w odpowiednim miejscu.
- Wpisy `count()` oznaczają, że podczas kompilacji, kompilator sam policzy ilość sekcji/obiektów w `.par` (jest to bardzo wygodne). Jeżeli chcemy na stałę wpisać wartość to możemy ją wpisać zamiast `count()` i kompilator to przepuści.

- Aby dodać obiekt - należy skopiować jakiś obiekt i wkleić do środka sekcji w odpowiednio wybrane miejsce i pozmieniać dane.
- W takiej kolejności w jakiej jest header i obiekty w danej sekcji, to w takiej kolejności zostanie skompilowana sekcja.
- Nazwy zmiennych mogą być dowolne, ale wyjątkami są nazwy `number_of_objects` oraz `number_of_sections` - te nazwy są zastrzeżone dla kompilatora.
- Program **PARim** czyli kompilator obsługuje komentarze także można je zostawiać w plikach.

5. Kompilacja plików .cpp do postaci .par przez PARim

PARim jest to program, a dokładnie kompilator służacy do kompilacji całego katalogu z plikami sekcji `.cpp` do postaci `.par`.

Jak używać?:

1. Włączamy program.
2. Wpisujemy nazwe katalogu.
3. Klikamy enter i za chwilkę powinno się zrobić.
4. Wyłączenie okna konsoli sygnalizuje zakończenie pracy programu.

Program działa również w trybie ARGC&ARGV. Przykład użycia:

```
PARim.exe <nazwa katalogu wejściowego>
```

Szybki Import:

Aby w szybki sposób kompilować katalog z plikami `.cpp`, należy korzystać z `IMPORT.ps1` lub `IMPORT.bat`. W celu modyfikacji argumentów/konfiguracji należy zedytować plik `.bat/.ps1` przez notatnik.

Przykład pliku .bat:

```
PARim.exe <nazwa katalogu wejściowego>
```

Przykład pliku .ps1:

```
.\PARim.exe .\<nazwa katalogu wejściowego>
```