

Parser Documentation 1.0

Contents

1. Introduction
2. Config file rules
3. Database interaction
4. Extra





Introduction

Welcome. This documentation is meant to help guide you how to use my parser.

At this point you should have these items:

- Program source code
- Program executable
- Provided default config file

Preliminaries: the program will look for a “**DefaultEngine.config**” file. If it’s not present the program will exit out immediately. If you’re using an executable, the folder structure should look like this:

 DefaultEngine.config	4/6/2020 11:25 PM	XML Configuration...	3 KB
 RegexHW.exe	4/6/2020 11:48 PM	Application	536 KB
 RegexHW.ilc	4/6/2020 11:48 PM	Incremental Linker...	1,670 KB
 RegexHW.pdb	4/6/2020 11:48 PM	Program Debug D...	1,668 KB

Also, if the file is empty - the program will exit early.

If you decide to use the already existing config file - it includes a bunch of tests to make sure the parser executes correctly. Thus, your initial output would look something like this:

```

C:\Users\denni\Source\Repos\RegexHW\Debug\RegexHW.exe
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 5. Reason: no whitespace in section definition is allowed
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 6. Reason: empty sections are not allowed
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 7. Reason: remove all # signs in the definition of a section
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 9. Reason: no whitespace in section definition is allowed
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 10. Reason: no whitespace in section definition is allowed
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 12. Reason: the subsection or section is empty
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 13. Reason: the subsection or section is empty
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 14. Reason: special characters are not allowed inside sections or subsections
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 16. Reason: the value that you defined has only 1 " signs. Please fix it. It could be because you used
the # sign as it denotes a comment
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 18. Reason: the value that you defined has only 1 " signs. Please fix it. It could be because you used
the # sign as it denotes a comment
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 19. Reason: the value that you defined has too many " signs. Please fix it
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 20. Reason: the value that you defined has only 1 " signs. Please fix it. It could be because you used
the # sign as it denotes a comment
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 28. Reason: no valid data type could have been deducted from the value
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 41. Reason: no valid data type could have been deducted from the value
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 42. Reason: no valid data type could have been deducted from the value
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 45. Reason: the given array is empty
[Tue Apr 7 10:51:29 2020] WARNING::String::Message: Invalid section on line: 47. data written: asdfasdfsadfsadafsga. The parser couldn't extract data from it. Please read the
documentation.
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 50. Reason: special characters besides ", . and ; are not allowed inside arrays
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 51. Reason: the array is incomplete. Please eliminate it
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 52. Reason: a malformed member of an array has been found. It's type cannot be deducted
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 53. Reason: a malformed member of an array has been found. It's type cannot be deducted
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 54. Reason: a malformed member of an array has been found. It's type cannot be deducted
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 55. Reason: the types inside the array are not the same
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 58. Reason: the key contains special characters that are not allowed. Please remove them
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 60. Reason: the key contains special characters that are not allowed. Please remove them
[Tue Apr 7 10:51:29 2020] ERROR::String::Message: Invalid section on line: 78. Reason: the section you specified for the subsection is out of order or doesn't match. Please reorg
anize
Parsing is done! DB is up! Commands available:

```

You can study the existing config file and some of the errors / warnings to understand better what the parser will and will not accept. For further details on the rules of the program please check the sections below.

Config file rules

The parser accepts only 3 types of data that would go into the database:

- [sections]
- [section:subsection]
- key=value

General:

- To create comments you can use the # sign
- Whitespace outside the parsed types will be ignored

Rules on sections and subsections:

- No whitespace inside the [] is allowed
- Empty sections (i.e [] or [:]) are invalid
- No special characters besides ":" are allowed inside the []
- Unfinished subsections ([:something] or [something:]) are also invalid
- Putting a # inside the section will cause the the right side of the section to be commented out, but the parser will detect it and throw an error

Rules on key=value pairs:

- Any whitespaces between the pair (i.e key = value) will be ignored

- Putting values like “#” will cause an error as the comment is treated above the string value due to consistency. Thus including a # sign is not allowed inside strings and the parser will detect it as an error
- Strings like this: “””” are not allowed due to the nesting issues. The parser will throw an error if it detects such a string
- Keys can’t include special characters (^,+” etc), but are allowed to include digits like this: key123=”something”
- Arrays also can’t include special character for parsing purposes besides “ as it denotes a string
- To enumerate members in the array you use ;. The parser will detect if you use an incorrect enumerator
- An array can hold only a single type of data. Mixing data types will cause the parser to throw an error

The order matters

When defining sections, sub sections and key=value pairs you should keep in mind the structure of the file.

- You cannot define a keyvalue pair unless there was a previously defined section or subsection
- You cannot define a subsection if there has not been defined a section that is used for the subsection. I.e:

[section]

[somethingelse:subsection]

Is invalid as instead of “somethingelse” it should be “section”

- Simply defining sub sections is also invalid unless there had been a section for it to go under.

Database interaction

To properly interact with the database I advise to follow the rules listed below.

- ListAllSections doesn’t require any params and is not case sensitive
- ListNamedSection requires a valid section parameter case sensitive
 - Besides giving a true/false section existence, it will list all of its subsections and key=value pairs (no key value pairs will be given in subsections though)
- ListSubsections requires a valid section parameter case sensitive

- ListKeyValuePairsInSection requires a valid section param or a subsection that is written like this: section:subsection
 - It is a special function for retrieving a list of key=value pairs under the specified section or subsection
- GetEntry requires a valid section param or a subsection param that is written like this: section:subsection and a valid key for an entry. The key is always the same as the key for the key=value pair
 - Params in get entry are comma separated. Example query:

CMD SECTION KEY

Please Enter Command: `getentry section:subsection, key`

- Upon calling this function the entry is loaded in, which allows you to do extra things to it:
 - GetKey simply retrieves the key from the pair
 - GetValue returns the value
 - GetType returns the type of the value

Some misc:

- END denotes the end of the list

Extra

If you encounter any bugs with the application feel free to submit an issue or a PR for a bugfix here: <https://github.com/DennisSSDev/SmorcEngineConfigParser>