Wet Pancake CLI 0.8.2 - ALPHA

Project Information

- Project URL: https://github.com/CraigOates/Wet-Pancake
- MyGet: https://www.myget.org/feed/the-immutable-null/package/nuget/WetPancake
- NuGet URL: T.B.A
- Written with .Net 4.7

Summary

Wet Pancake CLI is a console program for generating text. Its main aim is to provide a direct way to interact with the Wet Pancake library/NuGet. If you want to include Wet Pancake (NuGet) in your software project, add it via the Package-Manager console with,

```
// MyGet Feed - replace the version number accordingly.
PM> Install-Package WetPancake -Version 0.8.0-alpha1 -Source https://www.myget.org/F/the-immutable-null/api/v3/index.json
// Expected NuGet Feed
// At the time of this release, Wet Pancake was not published on NuGet.
PM> Install-Package WetPancake -Version 0.8.0-alpha1
```

Command-Methods

The parameters used in Wet Pancake CLI operate within the following ranges,

- Gibberish Level: 2-20
- File Type: plain-text (.txt)
- Sentences: greater then 0

If you are not using one of the built-in template files, your .txt file must contain at least one **".", "!" or "?"**. It does not matter which one is included as long as there is one. If you are unsure, use the `ValidFile` command-method to confirm you .txt file is compatible with Wet Pancake CLI.

Test

Prints a test message to the console.

Example: `> Test`

Help

Lists out the commands this program offers.

Example: `> Help`

Exit

Exits out of the program.

Example: `> Exit`

RequestAllTemplateFiles

Returns a list of all the available .txt files built-in to Wet Pancake.

Example: `> RequestAllTemplateFiles`

CleanText

(sentences: int) (text: string)

Checks to see if the string matches the desired sentence count and removes any over that limit. If the string has less sentences than the number requested, it will not change. Sentences must be greater than 0 and text must contain at least 1 "." "?" "!"

Example: `> CleanText 1 "This is a test sentence. And, this one needs to be removed."`

ValidateFile

(filePath: string)

Checks the text in the specified .txt file to see if it is compatible with this program. For a file to be compatible, it must be a .txt file and contain at least one '.', '!' or '?'.

Example: `> ValidateFile "C:/your-file.txt"`

GenerateCleanText

(gibberish-level: int) (sentences: int) (copy-to-clipboard: bool)

Generates text using the gibberish-level and number of sentences specified by the user. The result goes through an extra "cleaning" process to remove any artefact sentences. Use this if you cannot tolerate the odd extra sentence. With that said, it does mean it is slower than its `GenerateText` counterpart. Gibberish-level must be between 2 and 20. Sentences must be greater than 0. Pass in "true" to copy the result straight to the clipboard. Pass in "false" or leave blank to not copy the result.

Examples:

- `> GenerateCleanText 5 10 true`
- `> GenerateCleanText 3 7 false`
- `> GenerateCleanText 9 12`

GenerateCleanTextFromFile

(gibberish-level: int) (sentences: int) (file path: string) (copy-to-clipboard: bool)

Loads the specified .txt file and generates text based on it, using the gibberish-level and number of sentences specified by the user. The result goes through an extra "cleaning" process to remove any artefact sentences. Use this if you cannot tolerate the odd extra sentence. With that said, it does mean it is slower than its `GenerateTextFromFile` counterpart. Gibberish-level must be between 2 and 20. Sentences must be greater than 0. Pass in "true" to copy the result straight to the clipboard. Pass in "false" or leave blank to not copy the result.

Examples:

- `> GenerateCleanTextFromFile 3 6 C:/yourfile.txt true`
- `> GenerateCleanTextFromFile 5 9 C:/yourfile.txt false`
- `> GenerateCleanTextFromFile 7 15 C:/ yourfile.txt`

GenerateRandomText

(copy-to-clipboard: bool)

Generates random text, the number of sentences generated is randomly determined. Pass in true to copy result straight to clipboard. Pass in "false" or leave blank to not copy the result.

Examples:

- `> GenerateRandomText true`
- `> GenerateRandomText false`
- `> GenerateRandomText`

GenerateText

(gibberish-level: int) (sentences: int) (copy-to-clipboard: bool)

Generates text using the gibberish-level and number of sentences specified by the user. This command does not run the result through the extra "cleaning" process like `GenerateCleanText`. This means this command is faster but it might produce an extra sentence on the odd occasion. Use this if you prefer speed over accuracy. Gibberish-level must be between 2 and 20. Sentences must be greater than 0. Pass in "true" to copy the result straight to the clipboard. Pass in "false" or leave blank to not copy the result.

Examples:

- `> GenerateText 5 10 true`
- `> GenerateText 7 15 false`
- `> GenerateText 10 5`

GenerateTextFromFile

(gibberish-level: int) (sentences: int) (file path: string)

Loads the specified .txt file and generates text based on it, using the gibberish-level and number of sentences specified by the user. This command does not run the result through the extra "cleaning" process like `GenerateCleanTextFromFile`. This means this command is faster but it might produce an extra sentence on the odd occasion. Use this if you prefer speed over accuracy. Gibberish-level must be between 2 and 20. Sentences must be greater than 0. Pass in "true" to copy the result straight to the clipboard. Pass in "false" or leave blank to not copy the result.

Examples:

- `> GenerateTextFromFile 3 6 C:/yourfile.txt true`
- `> GenerateTextFromFile 6 13 C:/yourfile.txt false`
- `> GenerateTextFromFile 12 8 C:/yourfile.txt`