**ACT Cooperative Learning Handbook**

**(Trial operation and construction version)**

**directory**

1. **Let's get to know the ACT**
2. **Conduct command-line learning**

2.0. Environmental Requirements

2.1.uPot!Command

2.2. Reint/f command

2.3.mT/t command

2.4.RIF command

2.5. Verinspect command

1. **Look for ACT grammar patterns**

**Let's get to know the ACT**

**ACT is a new programming language in trial operation, with concise syntax and cross-platform portability. ACT is a Mercurial command-line application in this release. The user operates with preset commands and conforms to the rules of syntax. ACT is written in Python and C and inherits its cross-platform nature.**

**ACT is always open source and free, open to all kinds of developers, and developers are free to improve and optimize ACT and contribute to the ACT project.**

**At present, ACT is the first version, only simple commands and can not achieve multi-line compilation, due to limited personal power, the update speed will be slow, I hope you understand! There are any loopholes in the source code, welcome to correct. For more information about ACT's version updates, please refer to the ACT Version Plan.**

**Thank you to all users and contributors!**

**2023.11.25**

**ACT,Mr.ye**

**Conduct command-line learning**

**Environmental Requirements:**

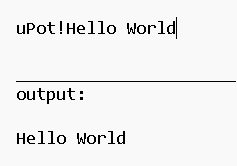
1. **Python 3.11 or later must be configured.**
2. **i3 2357M and its equivalent processors.**
3. **The storage free area is greater than 20MB.**
4. **DDR3 1066/1333 and above memory.**
5. **Windows 7 or later**

**Conduct command-line learning**

**uPot! command:**

**Use the uPot! command to declare the output instruction to the compiler, and type it after "!". The content of the output. uPot = output。**

**Example:**

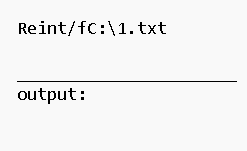


**Conduct command-line learning**

**Reint/f command:**

**ExploitThe reint/f command declares the deletion directive to the compiler, and after "f", type the path of the file to be deleted. Reint = Retreat into。**

**Example:**

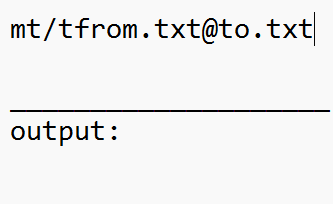


**Conduct command-line learning**

**mT/t@ command:**

**Use the mT/t command to declare the move command to the compiler, and type the full file address of the file to be moved after "t", and then type the full file address to be moved to after @. mT = move to。**

**Example:**

****

**Conduct command-line learning**

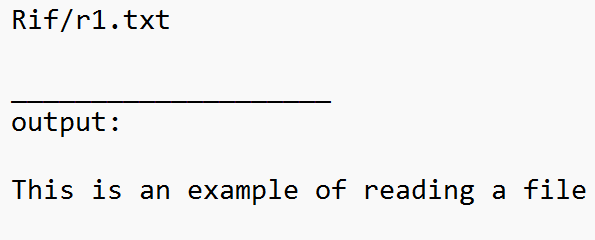
**Rif Command:**

**Use the Rif command to declare a write or read command to the compiler, and type the operation type after "f".’/r’Read or’/w’Write, when typing’/r’, then type the path to the file being manipulated;’/w’, then type’@’，’@’then type what you need to write.**

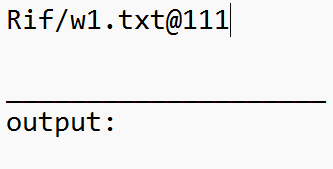
**Rif = write for**

**Example:**

Read:

****

Write:

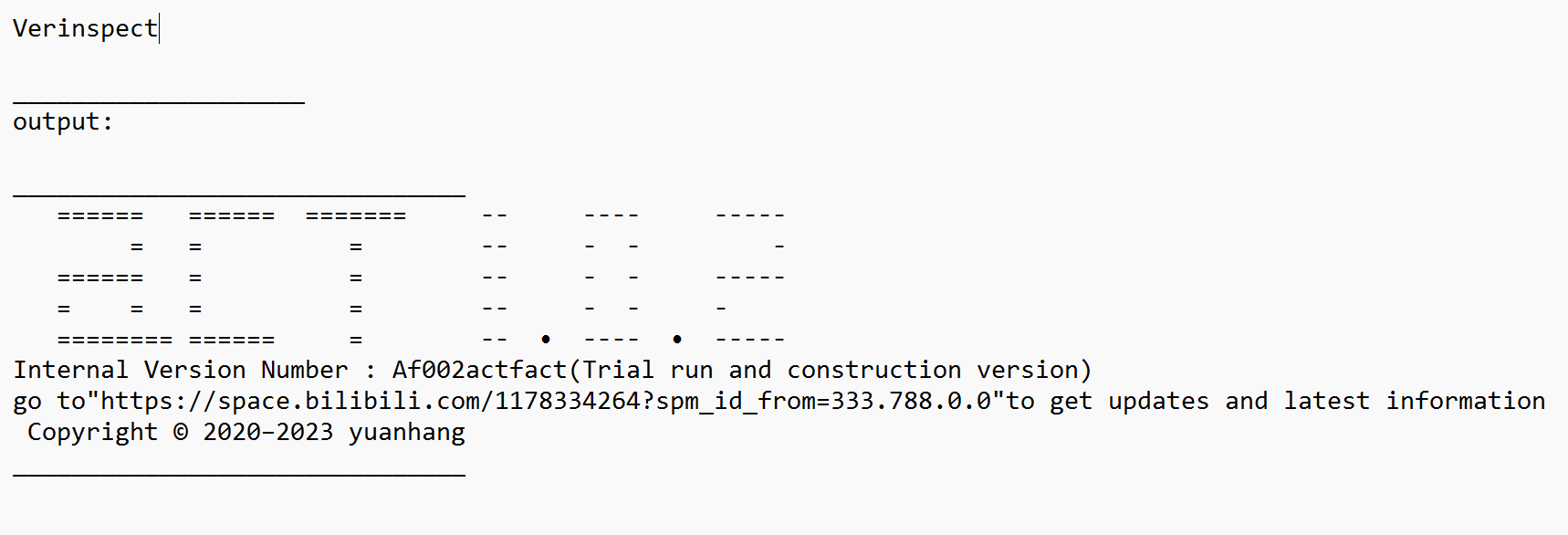


**Conduct command-line learning**

**Verinspect command:**

**Use the Verinspect command to query information such as the version number of the compiler.**

**Example:**

****

**Look for ACT grammar patterns**

**The naming convention for ACT commands mostly follows the capitalization of the second letter, and when the second letter is a vowel letter, the first letter is capitalized instead.**

**At this stage, ACT does not distinguish between data types, and starts with "!" As a separator, generally type a value after "!". When there is a purposeful relationship between the front and back parameters, @ is used as the separator. The "/" is usually a separator between the command and the name of the operation contained in the command.**

**The "t" in ACT is usually used as the "to"; f" is usually used as for; "r" is usually read as "read"; w" is usually used as a write.**

****