TATA CONSULTANCY SERVICES
Experience certainty.

Confidentiality Statement

This document contains confidential information of Tata Consultancy Services Limited, which is provided for the sole purpose of permitting the recipient to evaluate the proposal submitted herewith. In consideration of receipt of this document, the recipient agrees to maintain such information in confidence and to not reproduce or otherwise disclose this information to any person outside the group directly responsible for evaluation of its contents, except that there is no obligation to maintain the confidentiality of any information which was known to the recipient prior to receipt of such information from Tata Consultancy Services Limited, or becomes publicly known through no fault of recipient, or is received without obligation of confidentiality from a third party owing no obligation of confidentiality to Tata Consultancy Services Limited.

Tata Code of Conduct

We, in our dealings, are self-regulated by a Code of Conduct as enshrined in the Tata Code of Conduct. We request your support in helping us adhere to the Code in letter and spirit. We request that any violation or potential violation of the Code by any person be promptly brought to the notice of the Local Ethics Counselor or the Principal Ethics Counselor or the CEO of TCS.   All communication received in this regard will be treated and kept as confidential.

TATA CONSULTANCY SERVICES

Experience certainty.

TATA

TATA

TATA CONSULTANCY SERVICES

Virtual ILP – Introduction to awk Filter

Content Manual Version 1.1

December 2014

(ILP Guwahati)

Table of Content

1. Introduction to AWK Filter

AWK Stands for 'Aho, Weinberger, and Kernighan'.

It derives its name from the first letter of the last name of its three authors namely Alfred V. Aho, Peter J. Weinberger and Brian W. Kernighan.

Awk is a scripting language which is used for processing or analyzing text files. Or we can say that awk is used for grouping of data based on either a column or field, or on a set of columns. Mainly it's used for reporting data in a useful manner.

It also employs Begin and End Blocks to process the data. It searches one or more files to see if they contain lines that match specified patterns and then perform associated actions. AWK is an advance filter tool.

2. Simple awk Filtering

Syntax of awk:

```
~$ awk 'pattern {action}' input-file
```

Let’s take a input file with the following data

```
~$cat awk_file
Name,Marks,Max_Marks
Peter,200,1000
Sam,500,1000
Greg,1000
Abharam,800,1000
Henry,600,1000
Peter,400,1000
```

Example: Default behavior of awk

Print all the lines from a file. By default, awk prints all lines of a file, so to print every line of above created file use below command:

```
~$ awk '{print}' awk_file
```

```
Name,Marks,Max_Marks
Peter,200,1000
Sam,500,1000
```

```
Greg,1000
Abharam,800,1000
Henry,600,1000
Peter,400,1000
```

Example 2: Print only specific field

Print filed 2nd & 3rd

```
~$ awk -F”,” {print $2,$3;}' awk_file
```

```
Marks Max_Marks
200 1000
500 1000
1000
800 1000
600 1000
400 1000
```

In the above command we have used the option -F “,” which specifies that comma (,) is the field separator in the file.

Example: Pattern Matching

Print the lines which matches the pattern (lines which contains the word “Henry & Peter”)

```
~$ awk '/Henry|Peter/'  awk_file
```

```
Peter,200,1000
Henry,600,1000
Peter,400,1000
```

3. *Initialization and Final Action*

BEGIN and **END** statements are help full in displaying information before and after executing actual AWK script.

BEGIN block is evaluated before awk starts processing the actual awk script; it’s an excellent place to initialize the FS (field separator) variable, print a heading, or initialize other global variables.

BEGIN block Uses:

- Declaring variables.
- Initialization variables for doing increment/decrements operations in main AWK code.
- Printing Headings/info before actual AWK code output.

END block is evaluated after all the lines in the awk script have been processed. Typically, the **END** block is used to perform final calculations or print summaries that should appear at the end of the output stream.

END block Uses:

- Printing final results, after doing operations in main AWK block.
- Printing Completion/info after actual AWK code output.

AWK tool is mainly used for reporting some data in useful manner. Without these BEGIN and END blocks the output will be meaningless.
Consider db.txt which contains below data:

```
Jones 2143 78 84 77
Gondrol 2321 56 58 45
RinRao 2122234 38 37
Edwin 253734 87 97 95
Dayan 24155 30 47
```

3.1. AWK BEGIN block

This is a block code which is executed before executing actual AWK script.

BEGIN block Syntax

awk ‘BEGIN{awk initializing code}{actual AWK code}’ filename.txt

Example: Print a meaning full info before actual AWK output.

```
~$ awk ‘BEGIN{print “#########################\nThis is the output of filtered data\n#########################”}{print $0}’ db.txt
```

Output:

```
###########################
This is the output of filtered data
###########################
Jones 2143 78 84 77
Gondrol 2321 56 58 45
RinRao 2122234 38 37
Edwin 253734 87 97 95
Dayan 24155 30 47
```

Example:

We can use even BEGIN block to initialize user defined variables in it.
Adding 3rd column for every iteration to the same value.
3rd column
78
56
21
87
55
I want to add these numbers and display every addition when it happens

```
~$ awk ‘BEGIN{s=0}{s=s+$3;print s}’ db.txt
```

Output:

```
78
134
155
242
297
```

We initialized s in BEGIN block and incremented column 3 values in actual AWK code block.

Example:

Adding all the numbers present in the 4th column and print the total value at the end.

```
~$ awk ‘BEGIN{s=0}{s=s+$3}END{print s}’ db.txt
```

Output:

```
297
```

There is a slight difference between above 2 Examples. We used END block to print the last sum instead of printing the sum for every iteration. In our next post we will see how to use AWK inbuilt variables in detail.

3.2. AWK END block

This is piece of block which is executed after executing all the AWK code. This is the last AWK code going to execute.

Example:

Print some meaning full info after processing AWK code.

```
~$ awk ‘{print $0} END {print “###########################\n Completed printing filtered data\n#########################”}’ db.txt
```

Output:

```
Jones 21 78 84 77
Gondrol 23 56 58 45
RinRao 25 21 38 37
Edwin 25 87 97 95
Dayan 24 55 30 47
##########################
Completed printing filtered data
##########################
```

3.3. Combining BEGIN and END block

Example:

Print some meaning full info before and after AWK code execution.

```
~$ awk ‘BEGIN{print “############################\n This is the output of filtered data\n############################”}{print $0}END{print “#########################\n Completed printing filtered data\n#########################”}’ db.txt
```

Output:

```
#########################
This is the output of filtered data
#########################
Jones 21 78 84 77
Gondrol 23 56 58 45
RinRao 25 21 38 37
Edwin 25 87 97 95
Dayan 24 55 30 47
########################
Completed printing filtered data
########################
```