STRING FUNCTIONS IN ABINITIO

Version 1.0

Ponmani Srinivasan(ponmani.srinivasan@tcs.com)

TATA CONSULTANCY SERVICES

Contents

Document Version Control 3
1. INTRODUCTION: 4
2. STRING FUNCTIONS IN ABINITIO: 5
3. EXPLANATION: 6
4. APPENDIX 12

Document Version Control

Version	Date	Changes
1.0	24/02/2014	First Release

1. INTRODUCTION:

The Ab Initio software is a Business Intelligence platform containing six data processing products:

- Co>Operating System,
- The Component Library,
- Graphical Development Environment,
- Enterprise Meta>Environment,
- Data Profiler
- Conduct>It.

It is a powerful graphical user interface-based parallel processing tool for ETL data management and analysis. Graphical Development Environment provides an intuitive graphical interface for editing and executing applications. The strength of Ab Initio-ETL is massively parallel processing which gives it capability of handling large volume of data.

2. STRING FUNCTIONS IN ABINITIO:

There are numerous string function in abinitio:Listed below are the few of them and their extensive use in various realtime scenarios.

- String_length
- String_filter
- String_lrtrim
- String_index
- String_rindex
- String_substring
- String_replace
- String_filter_out
- Re_get_match
- Re_replace
- Re-split
- String_like
- String_repad
- String_join
- String_lpad
- String_prefix
- String_suffix
- String_is_alphabetic
- String_is_numeric
- Re_get_range_matches

3. EXPLANATION:

3.1.Objective: To Count the number of occurrence of spaces in a string

Input:”abc def ghi”

Function Used:string_length(string_filter("substring1 substring2 substring3"," "))

Output:2

Explanation: This function will count the number of spaces within a string

3.2.Objective: To split a string into substrings separated by whitespace

Input: Jack K Frencho”

Function Used:

```
begin
let string("")field_name=string_lrtrim(in.field_name);
let integer("")end_of_first_name=string_index(field_name," ");
let integer("")beginning_of_last_name=string_rindex(field_name," ");
out.last_name::string_substring(field_name,beginning_of_last_name+1,length_of(field_name));
out.first_name::string_substring(field_name,1,end_of_first_name-1);
        out.mid_name::string_substring(field_name,end_of_first_name+1,(length_of(field_name)
-  (end_of_first_name+beginning_of_last_name)));
end
```

Output: FirstName:Jack
MidName:K
LastName:Frencho

Explanation: This function will split a string into substrings separated by whitespaces.

3.3. Objective: To split a string into substrings separated by comma.

Input: CLEVELAND, OH ,44113

Function Used:

City:string_replace((string_substring(CLEVELAND, OH ,44113,1,(string_index(CLEVELAND, OH ,44113,","))))","," ")

State:string_filter_out(string_replace((string_substring(CLEVELAND, OH ,44113,(string_index(CLEVELAND, OH ,44113,",")),20)),",","),"0123456789")

ZipCode: string_substring((string_filter(CLEVELAND, OH ,44113,"0123456789")),1,5)

Output:

City:CLEVELAND
State:OH
ZipCode:44113

Explanation: To split the **Input** string into three substrings-City,State and ZipCode separately and to fetch only first 5 digits for ZipCode

3.4.Objective: To get the index of the first character of a substring of a string that matches a specified regular expression.

Input:" FBO Hines 333 West Wacker Drive 456 LP"

Function Used: re_index("FBO Hines 333 West Wacker Drive 456 LP", "[0-9]+")

Output:10

Explanation: This function will return the index of first occurrence of numeric value

3.5.Objective:To get the first substring in a string that matches a regular expression.

Input: :" FBO Hines 333 West Wacker Drive 456 LP"

Function Used:re_get_match("FBO Hines 333 West Wacker Drive 456 LP", "[0-9]+")

Output:333

Explanation: This function will return the first substring which matches the numeric pattern [0-9]

3.6.Objective: To replace all substrings in a string that match a specified regular expression.

Input: "2800 Post Oak Boulevard, 30th street Suite 5000"

Function Used: re_replace("2800 Post Oak Boulevard, 30th street Suite 5000", "[0-9]+", "[No &]")

Output:No 2800 Post Oak Boulevard, No 30 th street Suite No 5000

Explanation: This function replaces numeric substrings of the string with the string "No &", where the matched substrings replace the ampersand character

3.7.Objective:To split a string into vector substring using a specified regular expression.

Input: CLEVELAND, OH , 44113

Function Used: re_split("CLEVELAND,OH,44113", ",")

Output:

[vector

"CLEVELAND",
"OH",

"44113"]

3.8.Objective:To Compare the contents of two strings,and return a string containing characters that appear in both of them.

Input:"CLEVELAND, OH 44113"

Function Used:string_filter("CLEVELAND, OH 44113 ","0123456789")

Output:44113

3.9.Objective:To compare two **Input** strings and returns characters that appear in one string but not in the other.

Input:"CLEVELAND, OH 44113"

Function Used:string_filter("CLEVELAND, OH 44113 ","0123456789")

Output:CLEVELAND,OH

3.10.Objective:To test whether a string matches a specified pattern.

Input:"CLEVELAND,OH 44113"

Function Used:string_like("CLEVELAND,OH 44113","cleveland%")

Output:0

(**Note:**string_like function is case-sensitive.And so the result of above function is 0.)

String_like(("CLEVELAND,OH 44113","CLEVELAND%)

Output:1

3.11.Objective: To return a string of a specified length trimmed of any leading and trailing blank characters, and then right-padded with a given character.

Input:"702 W. HAMILTON ST"

Function Used:string_repad(:"702 W. HAMILTON ST",21,"REET")

Output: 702 W. HAMILTON STREET

Explanation: This function right-pads the string "702 W. HAMILTON ST " with "REET", returning a string of length 21

3.12.Objective:To concatenate vector string elements into a single string.

Input:"CLEVELAND","OH","44113"

Function Used:string_join([vector "CLEVELAND","OH","44113"],",")

Output:CLEVELAND,OH,44113

Explanation: This function will combine the vector elements separated by comma(,)

3.13.Objective: To Return a string of a specified length, left-padded with a given character.

Input:"702 W. HAMILTON STREET"

Function Used:string_lpad("702 W. HAMILTON STREET",23,"No")

Output: No 702 W. HAMILTON STREET

Explanation: This function left-pads the string "702 W. HAMILTON STREET " with “No”, returning a string of length 23

3.14.Objective:To Return a substring that starts at the beginning of the string till a specified length.

Input:”50PUBLICSQUARE, SUITE 1150”

Function Used:string_prefix(”50PUBLICSQUARE, SUITE 1150”,14)

Output: 50PUBLICSQUARE

3.15.Objective:To Return a substring of a specified length that ends at the end of the string

Input:”50PUBLICSQUARE, SUITE 1150”

Function Used:string_suffix(”50PUBLICSQUARE, SUITE 1150”,10)

Output: SUITE 1150

3.16.Objective:To check whether a string starts with an alphabet.

Input:”FORESTAR (USA) REAL ESTATE GROUP”

Function Used:string_is_alphabetic(” FORESTAR (USA) REAL ESTATE GROUP”)

Output:1

Explanation: This function returns 1 as the string starts with an alphabet

3.17.Objective: To check whether a string starts with numeric.

Scenario: To check for the occurrence of numeric value to mark the start of address

Input:” 6300 Bee Cave Road ”

Function Used:string_is_numeric(”6300 Bee Cave Road”)

Explanation: This function returns 1 as the string starts with numeric

Output:1

3.18.Objective: To return a vector that describes the index and length of a string that matches a specified regular expression

Input:"CLEVELAND,OH,43114"

Function Used:re_get_range_matches("CLEVELAND,OH,43114", "[0-9]+")

Scenario: To count the length of numeric value(ZipCode).To check if Zipcode is of five digits. This function will return 5

Output: [vector
[record
index 14
length 5]

4. APPENDIX

On Web:

- abinitio-basic-tutorial.weebly.com/
- ab-initio-tutorials.blogspot.com/
- abinitio-tutor.blogspot.com/2013/10/string-functions-abinitio-tutor.htmlCached