Regex Custom Function for FileMaker

Version 1.0.0

Document Version By Jangwon Lee

Completed on March 19th, 2019

Doc #	Doc Author	D. Write DT	Code #	Code Author	Code DT
1.0.0	Jangwon Lee	3-19-2019	N/A	N/A	N/A

Introduction

This document will cover the use case and specifications of RegEx custom function for FileMaker. It will also cover the supported RegEx expressions and custom code specifications. Upon reading the document you should have a clear understanding of what the project is to be used for , how the custom code is structured, and how to install it to your project (to be specified).

Project Info

RegEx, or regular expressions, is a form of text expression used for advanced text searching. It uses special characters and structure to make dynamic text searching. For instance, by using the text expression “\w{5}”, you will find all instances in a specified text with 5 consecutive alphanumeric characters.

By creating RegEx as custom function on FileMaker, you can call regular expressions from scripts regardless of the environment used to work on it.

Use Cases

Having ReGex functions that don’t need external plugins is highly useful for sharing. It is also necessary for searching because there isn’t any RegEx implementation that can search with indexing.

RegEx is useful for any application that uses dynamic searching, or deals with messy text data. One example use case is that of extracting credit card number from existing text data. By using regex you can extract finite length of arbitrary digits with multiple possible spacing (xxxx-xxxx-xxxx-xxxx, xxxxxxxxxxxxxxxx, xxxx xxxx xxxx xxxx, xxxx-xxxx-xxxx-xxxx) without prior knowledge of credit card number.

Supported Regular Expressions

Char	**Function**
.	Matches any character (wildcard)
^	Matches beginning of string
$	Matches end of string
*	Matches preceding pattern zero or more times (greedy)
+	Matches preceding pattern one or more times (greedy)
?	Matches preceding pattern zero or one time (greedy)
\s	Matches whitespace (space, \t, \f, \r, \n, \v)
\S	Matches non-whitespace
\w	Matches alphanumeric character
\W	Matches non-alphanumeric character
\d	Matches digit
\D	Matches non-digit

The custom function also supports escaped characters (all whitespaces, backslash, …).

Installation Guide (to be specified)

Though by nature a custom function is bound to the project on which it is developed, these functions are meant to be used like an api. The RegEx custom function can be run on FileMaker Pro Advanced 17 and above, with possible compatibility with previous and lower versions.

Functions

Public Functions

RE_MatchFirst (text, pattern)

Returns the first instance in text that matches the regular expression pattern.

Function Info:
- Return type: text
- Input type for pattern: text (regular expression)
- Input type for text: text

RE_MatchFirst_Position (text, pattern)

Returns the position of the first instance in text that matches the regular expression pattern. NOT IMPLEMENTED YET

Function Info:

- Return type: number
- Input type for pattern: text (regular expression)
- Input type for text: text

RE_MatchAll (text, pattern)

Returns all instances of text that matches the regular expression pattern. NOT IMPLEMENTED YET.

Function Info:
- Return type: List<text>

- Input type for pattern: text (regular expression)

- Input type for text: text

RE_MatchAll_Positions (text, pattern)

Returns the positions of all the instances of text that matches the regular expression pattern. NOT IMPLEMENTED YET.

Function Info:
- Return type: List<number>

- Input type for pattern: text (regular expression)

- Input type for text: text

RE_MatchOne (text, pattern)

Returns a random instance of text that matches the regular expression pattern. NOT IMPLEMENTED YET.

Function Info:
- Return type: text
- Input type for pattern: text (regular expression)

Input type for text: text

RE_MatchOne_Position (text, pattern)

Returns the position of a random instance of text that matches the regular expression pattern. To get the matching value, split and get right of text at position index and call RE_MatchFirst function. NOT IMPLEMENTED YET.

Function Info:

- Return type: number
- Input type for pattern: text (regular expression)
Input type for text: text

Pseudo-Private Level Zero Functions

The functions here aren’t meant to be called by the user, but is left accessible. These functions don’t call any regular expression custom functions, and are the lowest level of implementation.

RE__DetectMetaCommand (_pattern)

Detects modifier command in text at specified index. Returns 0 if there’s no command, 1 if zero or more character needs to be recognized (*), 2 if one or more character needs to be recognized (+), and 3 if zero or one characters need to be recognized (?). Used by StartsMetaSingle, StarstMetaSlash, and StartsSquareBracket functions at index+increment. Is greedy; will match as many characters as possible.

Function Info:

- Return type: int; 0 (no command), 1 (zero or more), 2 (one or more), 3 (zero or one)

- Input type for _pattern: text (regular expression)

RE__ZeroAllowed (_metaCommand)

Checks if meta command (0, 1, 2, 3) allows zero instances of detection. Returns True if meta command is 1 or 3 (“zero or more” and “zero or one” respectively) and False if meta command is 0 or 2 (“one” and “one or more” respectively).

Function Info:

- Return type: boolean

- Input type for_metaCommand: int (0-3 inclusive)

RE__SearchMore (_metaCommand)

Checks if meta command (0, 1, 2, 3) should perform greedy search.

Checks if meta command (0, 1, 2, 3) allows zero instances of detection. Returns True if meta command is 1 or 2 (“zero or more” and “one or more” respectively) and False if meta command is 0 or 3 (“one” and “zero or one” respectively).

Function Info:

- Return type: boolean

- Input type for_metaCommand: int (0-3 inclusive)

Pseudo-Private Level One Functions

The functions here aren’t meant to be called by the user, but is left accessible. These functions call custom functions that are level zero or below.

RE__StartsAlpha (_text, _command)

Searches for alphabet characters at start of text. Returns -1 if failed, 0 if command doesn’t increment text, 1 if only one instance is needed, and positive integer for increment.

Function Info:

- Return type: int; -1 (failed), 0 (no increment), 1 (single increment), …

- Input type for _text: text

- Input type for _command: int; 0 (no command), 1 (0-n), 2 (1-n), 3 (0-1)

RE__StartsAlphaNum (_text, _command)

Searches for alphanumeric characters at start of text. Returns -1 if failed, 0 if command doesn’t increment text, 1 if only one instance is needed, and positive integer for increment.

Function Info:

- Return type: int; -1 (failed), 0 (no increment), 1 (single increment), …

- Input type for _text: text

- Input type for _command: int; 0 (no command), 1 (0-n), 2 (1-n), 3 (0-1)

RE__StartsDigit (_text)

Searches for numeric characters at start of text. Returns -1 if failed, 0 if command doesn't increment text, 1 if only one instance is needed, and positive integer for increment.

Function Info:

- Return type: int; -1 (failed), 0 (no increment), 1 (single increment), …
- Input type for _text: text
- Input type for _command: int; 0 (no command), 1 (0-n), 2 (1-n), 3 (0-1)

Function Info:

- Return type: 0 or 1 (as text)
- Input type for text: text

RE__StartsWhitespace (_text)

Searches for whitespace characters at start of text. This includes new lines, tabs, spaces, and more. Returns -1 if failed, 0 if command doesn't increment text, 1 if only one instance is needed, and positive integer for increment.

Function Info:

- Return type: int; -1 (failed), 0 (no increment), 1 (single increment), …
- Input type for _text: text
- Input type for _command: int; 0 (no command), 1 (0-n), 2 (1-n), 3 (0-1)

Pseudo-Private Level Two Functions

The functions here aren't meant to be called by the user, but is left accessible. These functions call custom functions that are level one or below.

RE__StartsMetaSingle (text, _pattern, _index)

Performs single-character RegEx operation on text at specified index. Returns -1 if failed, 0 if command doesn't increment text, and positive integer for increment. Calls StartsMetaCommand function.

RE__StartsMetaSlash (text, _pattern, _index)

Performs slash-character RegEx operation on text at specified index. Returns -1 if failed, 0 if command doesn't increment text, and positive integer for increment. Calls StartsMetaCommand function.

Pseudo-Private Level Three Functions

The functions here aren't meant to be called by the user, but is left accessible. These functions call custom functions that are level three or below.

Challenges

The filemaker environment provides only limited programming control. The way custom function works is that you can't specify a class in which you can store all the functions and variable. Any sub-function to be used privately by bigger function cannot be hidden away, and any value to be used has to be specified as a parameter. Everything about the code structure calls for functional programming as opposed to OOP.

As of writing, it is unknown whether it is possible to return or even use lists. There are api specification for list creation, but documentation and examples are limited in general. If not possible, regex command will only detect first instance of the string. It's possible to get next instance by getting the text on the right side of the result.

The current implementation of meta commands are all greedy; they will match as many characters as possible. This could lead to regex not detecting text that should otherwise match. Is intended to be changed into non-greedy implementation in later versions.

There are regex commands that aren't supported yet. The project is intended to be built as modular as possible so that new commands can be added without major changes to the existing code. New commands are intended to be added in later versions.