

Passing named values using a Script Parameter or a Script Result

Version 5, Modified 2018-10-09

There are many methods for passing name-value pairs from one FileMaker file to another.

This document describes a method (based on **six.fried.rice**) that is:

- Robust
- Backward compatible (as far back as FileMaker 8)
- Popular in the community
- Easy to understand (fits on one page!)

Suite of Custom Functions	Function Name	Parameters
	#	a_name ; a_value
	#decode	encoded_string
	#encode	unencoded_string
	#g	the_dictionary ; a_name
	#p	a_name
	#r	a_name

Name-value pairs are text... use GetAsDate (text), GetAsNumber (text), etc to cast type.

Sometime we use #p.d, #p.n, etc. for this.

Example usage in scripts:

```

Make New Activity_BTN
1
2 Set Variable [ $nameFirst ; Value: Contact::NameFirst ]
3 Set Variable [ $nameLast ; Value: Contact::NameLast ]
4 Set Variable [ $_ID_Contact ; Value: Contact::_ID ]
5
6 # Build the SFR dictionary
7 Set Variable [ $d ; Value: # ( "nameFirst" ; $nameFirst ) & "¶" & # ( "nameLast" ; $nameLast ) & "¶" & # ( "_ID_Contact" ; $_ID_Contact )
8 Perform Script [ "_New Activity_P" from file: "Activity" ; Parameter: $d ]
    
```

```

Name: $d
Value: # ( "nameFirst" ; $nameFirst ) & "¶" & # ( "nameLast" ; $nameLast ) & "¶" & # ( "_ID_Contact" ; $_ID_Contact )
    
```

Sample Dictionary with 3 name=value pairs
 <:nameFirst:=Tony Test:>
 <:nameLast:=White:>
 <:_ID_Contact:=123456:>

```

_New Activity_P
1
2 # Get SFR values from Script Parameter
3 Set Variable [ $nameFirst ; Value: #p ( "nameFirst" ) ]
4 Set Variable [ $nameLast ; Value: #p ( "nameLast" ) ]
5 Set Variable [ $_ID_Contact ; Value: #p ( "_ID_Contact" ) ]
6
7 # Do something with the values
    
```

2 Get named value from Script Param, Result, or any dictionary

```

#p ( a_name ) =
// Gets a Value from a Script Parameter 'Dictionary' based on a_name
// more notes in Custom Function named #

#r ( a_name ) =
// Gets a Value from a Script Result 'Dictionary' based on a_name
// more notes in Custom Function named #
    
```

Usage: #p ("nameFirst")
 Result: Tony Test

```

#g ( the_dictionary ; a_name ) =
// Gets a Value from a Dictionary based on a_name
// more notes in Custom Function named #

Let (
[
~a_name.encoded = "<" & #encode ( a_name ) & "=" ;
~a_name.encoded.start.position = Position ( the_dictionary ; ~a_name.encoded ; 1 ; 1 ) ;
~value.encoded.start.position = ~a_name.encoded.start.position + Length ( ~a_name.encoded ) ;
~value.encoded.end.position = Position ( the_dictionary ; ">" ; ~value.encoded.start.position ; 1 ) ;
~value.encoded.length = If ( ~value.encoded.end.position = 0 ; 999999999 ; ~value.encoded.end.position - ~value.encoded.start.position ) ;
~value.encoded = Middle ( the_dictionary ; ~value.encoded.start.position ; ~value.encoded.length ) ;
~a_value = If ( ~a_name.encoded.start.position = 0 ; "" ; #decode ( ~value.encoded ) ) ;
] ;
~a_value // return
)
    
```

```

#encode ( unencoded_string ) =
// builds an encoded name, value pair
// more notes in Custom Function named #
Substitute ( unencoded_string ;
[";" ; "/:" ] ;
["=" ; "/" ] ;
[">" ; ">" ] ;
["<" ; "<" ] ;
)

#decode ( encoded_string ) =
// decodes an encoded name, value pair
// more notes in Custom Function named #
Substitute ( encoded_string ;
["/:" ; ";" ] ;
["/" ; "=" ] ;
[">" ; ">" ] ;
["<" ; "<" ] ;
)
    
```

Internals

1 Build Encoded name=value pair

```

# ( a_name ; a_value ) =
// based on http://sixfriedrice.com/wp/passing-multiple-parameters-to-scripts-advanced/ // 2007-07-12
// and http://sixfriedrice.com/wp/filemaker-dictionary-functions/ // 2007-11-23
// minor readability changes by Tony White Designs, Inc.
// builds an encoded name, value pair

"<" &
#encode ( a_name ) &
"=" &
#encode ( a_value ) &
">"
    
```

Usage: # ("nameFirst" ; "Tony Test")
 Result: <:nameFirst:=Tony Test:>

fx Makes use of these FileMaker functions:

- Get (ScriptParameter)
- Get (ScriptResult)
- If (test ; result1 ; result2)
- Length (field)
- Middle (text ; start ; numberOfCharacters)
- Position (text ; searchString ; start ; occurrence)
- Substitute (text ; searchString ; replaceString)

