

Custom Functions

By leveraging [expr-eval](#) JavaScript library, now we can execute rules and evaluate expressions with multiple conditions

concat(values)

Description	:	Concatenates values
Alias	:	--
Returns	:	concatenated values
Example	:	<pre>concat(vo(123),',',' ',vo(124),' is correct') => a, b is correct</pre>

concatWithDelimiter(delimiter, values)

Description	:	Concatenates values with delimiter(Ignores empty or null strings)
Alias	:	cwd
Returns	:	concatenated values
Example	:	<pre>cwd(',',' ',vo(123),vo(124)) => a, b</pre>

getElementById(id)

Description	:	Gives element by form-fill-metadata-id
Alias	:	elbyid(id)
Returns	:	element of given id
Example	:	<pre>elbyid(123)</pre>

valOf(id)

Description	:	Gives the value of element by form-fill-metadata-id
Alias	:	vo(id)
Returns	:	value of given id
Example	:	<pre>vo(123)</pre>

enable(condition, destIds)

Description	:	Enables elements of destIds if condition evaluates to true
Alias	:	en(condition, destIds)
Returns	:	condition
Example	:	<pre>en(vo(121)=='a', 123)</pre>

disable(condition, destIds)

Description : Disables elements of destIds if condition evaluates to true
Alias : `dis(condition, destIds)`
Returns : `condition`
Example : `dis(vo(121)=='a', 123, 124)`

makeMandatory(condition, destIds)

Description : Makes elements of destIds mandatory if condition evaluates to true
Alias : `mm(condition, destIds)`
Returns : `condition`
Example : `mm(vo(121)=='a', 123, 124)`

makeNonMandatory(condition, destIds)

Description : Makes elements of destIds non-mandatory if condition evaluates to true
Alias : `mmn(condition, destIds)`
Returns : `condition`
Example : `mmn(vo(121)=='a', 123, 124)`

makeVisible(condition, destIds)

Description : Makes elements of destIds visible if condition evaluates to true
Alias : `mvi(condition, destIds)`
Returns : `condition`
Example : `mvi(vo(121)=='a', 123, 124)`

makeInvisible(condition, destIds)

Description : Makes elements of destIds invisible if condition evaluates to true
Alias : `minvi(condition, destIds)`
Returns : `condition`
Example : `minvi(vo(121)=='a', 123, 124)`

copyToFillData(condition, srcValue, destIds)

Description : Copies srcValue to destIds if condition evaluates to true
Alias : `ctfd(condition, srcValue, destIds)`
Returns : `condition`

Example :
 1) `ctfd(vo(121)=='a','SomeValue',123,124)`
 2) `ctfd(vo(121)=='a',elbyid(120),123,124)`
 3) `ctfd(true,concat(123,124))`

executeRuleById(condition, formFillMetadataIds)

Description : Executes all rules of other form-fill-data(to avoid defining same rules)
Alias : `erbyid(condition, formFillMetadataIds)`
Returns : `condition`
Example : `erbyid(true,121)`

clearField(condition, destIds)

Description : Clears destId fields if condition evaluates to true
Alias : `cf(condition, destIds)`
Returns : `condition`
Example : `cf(true,121,124)`

partyType()

Description : Gives the current partyType;
Alias : `pt()`
Returns : FP or SP or undefined
Example : `pt()`

memberType()

Description : Gives the current transaction memberType;
Alias : `mt()`
Returns : `CREATED_BY, CREATED_FOR, APPROVER, GENERIC_PARTY`
Example :
`mv(mt()=='APPROVER',340);`
`// this makes filldata-340 visible if the current transaction member is approver`

addError(condition, message, ...destIds)

Description : Adds error message to destIds if condition evaluates to true
Alias : `adderr()`
Returns : `condition`
Example : `adderr(true,'Value is invalid',340)`

removeError(condition, formFillMetadataIds)

Description	:	Removes error message from destIds if condition evaluates to true
Alias	:	<code>remerr()</code>
Returns	:	condition
Example	:	<code>remerr(true, 340)</code>

formTag()

Description	:	Gives name of form field
Alias	:	<code>ft()</code>
Returns	:	name of field
Example	:	<code>ft(340);</code>

toLowerCase()

Description	:	Transforms input value to lowercase
Alias	:	<code>tolc()</code>
Returns	:	value transformed to lowercase
Example	:	<code>tolc(vo(340));</code>

toUpperCase()

Description	:	Transforms input value to uppercase
Alias	:	<code>touc()</code>
Returns	:	value transformed to uppercase
Example	:	<code>touc(vo(340));</code>

contains(sequenceToSearch,value)

Description	:	Returns true if and only if this string contains the specified sequence of char values.
Alias	:	<code>cntns()</code>
Returns	:	true or false
Example	:	<code>cntns('+','start');// false</code>

validationStatusOf(id)

Description	:	Returns validation status of the form-fill-data
Alias	:	<code>vso()</code>
Returns	:	true or false
Example	:	<code>vso(12345);// 'SUCCESS'</code>
Values	:	INIT, SUCCESS, FAIL, PENDING, ABORTED, TIMEOUT, WARN

sessionBasedMakeVisible(condition,param,destIds)

Description : Makes elements of destIds visible if condition evaluates to true and execute rule based on session based param

Alias : sbmvi(condition,param,destIds)

Returns : condition

Example : mvi(vo(121)=='a','GENERIC_STATIC',123,124)

sessionBasedMakeInvisible(condition,param,destIds)

Description : Makes elements of destIds invisible if condition evaluates to true and execute rule based on session based param

Alias : minvi(condition,param,destIds)

Returns : condition

Example : sbminvi(vo(121)=='a','SECOND_PARTY',123,124)

replaceRange(strng, start, end, subst, replaceEachChar = true)

Description : Replaces range of chars of a string with a substitute and replaceEachChar(optional, default=true) setting.

Alias : rplrng

Returns : string with range of chars replaced with subst

Example :

```
rplrng("1234567890ab",0,8,"X")
=> XXXXXXXX90ab
rplrng("1234567890ab",0,8,"X",false)
=> X90ab
rplrng("1234567890ab",0,8,"",false)
=> 90ab
```

regexTest(value, regex)

Description : Tests if a string matches the provided regular expression..

Alias : rgxtst

Returns : Returns true if the regular expression matches, otherwise returns false.

Example :

```
regexTest(vo(12345),'/^\\d{10}$/')
=> true or false depending on whether the value
of element 12345 is a 10-digit number.
```

```
regexTest(vo(12345),vo(12346))  
=> true or false depending on whether the value  
of element 12345 matches the regex provided by  
the value of element 12346
```

Pre-defined Functions

<https://github.com/silentmatt/expr-eval#pre-defined-functions>

Expression Syntax

<https://github.com/silentmatt/expr-eval#expression-syntax>

Usage

Ex1 :

Expression

```
minvi ( (vo (340)=="1" and  
vo (341)=="abc" ) , 342 ) ;mvi ( (vo (340)!="1" or  
vo (341)!="abc" ) , 342 )
```

Above expression makes filldata-342 invisible if value of filldata-340 is '1' and value of filldata-341 is 'abc'; And, makes filldata-342 visible if value of filldata-340 is not '1' or value of filldata-341 is not 'abc';