Virtual functions

See if the function is available for ConfiForms CLOUD - CLOUD

Important! Function names and field names are **CASE SENSITIVE**. It is very important to follow the correct letter casing as per documentation

In order to make it easier to integrate with other external systems, such as JIRA, for example, or transform values as you like and need we have implemented so called "virtual" functions, what you can call on field values.

Usage:

- When using in **IFTTT macro body or in ListView/PlainView/CalendarView** - the notation would be: **[entry.fieldname.virtual_function]** Please note that accessing values via [entry.field_name] notations brings you internal values.. which means you will need to make sure they are what you need - formatting dates via formatDate, accessing .label properties for choice-based fields and things like that.
- When using in **ConfiForms Field macro** then reference it by a field name, adding the virtual function name: **fieldname.virtual_function**

Useful, when you try to prepare a JSON or some other format when used together with IFTTT macro to enable integrations with other systems

Functions support ***chaining*** ! That means you can apply function on a result of the previous function as much as you like

See also Accessing field values and properties. You can use complex properties in your filters. For example filtering dropdown fields by values and by labels, filtering page type fields by page metadata fields, filtering user fields by, for example - email property

As of now, the following functions are supported:

CORE - means the function is available from ConfiForms 1.x

Function	Description	Using in ConfiForms Field macro	
`urlencode` CORE CLOUD	Does URL encode on given value, see "urlEncode" method in https://developer.atlassian.com/static/javadoc/confluence/4.0/reference/com/atlassian/confluence/util/GeneralUtil.html	myfield. urlencode	[
`escapeXML` CORE CLOUD	Escapes XML on given value, see "escapeXml" method in https://developer.atlassian.com/static/javadoc/confluence/4.0/reference/com/atlassian/confluence/util/GeneralUtil.html	myfield. escapeXml	[

`escape` CORE CLOUD	Escapes string as in "escapeForHtmlAttribute" method in https://developer.atlassian.com/static/javadoc/confluence/4.0/reference/com/atlassian/confluence/util/GeneralUtil.html	myfield. escapeForHtml Attribute	[e
`escapeJavaScript` CORE CLOUD	Escapes JavaScript from the value		
formatDate CORE CLOUD FROM V. 2.12.5 You can specify date format **and** timezone. When you dont specify the timezone a user's timezone is used to format the date/time value Example: formatDate(MM-dd-yyyy, UTC) Supported timezone arguments	Tries to format date fields in the specified format, expects date format as in http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html	myfield. format Date (JAVA_FORMAT)	[((
convertDate (FORMAT, TIMEZONE) FROM V. 2.13.5 CLOUD Supported timezone arguments	This function converts a timestamp (UTC) into the formatted date in a specified timezone Format can be anything supported by http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html Timezone argument can be anything documented here: Supported timezone arguments	Example: myfield. convertDate (yyyy-MM-dd'T'HH:mm:ss.SSS'Z', Europe/Tallinn)	
`jiraDate` CORE CLOUD FROM V. 2.12.5 You can specify date a timezone jiraDate(America/Chicago) Supported timezone arguments	Same as "formatDate" method above, but specifies constant DateFormat pattern (yyyy-MM-dd), suitable for JIRA REST API	myfield. jiraDate	[

`jiraDateTime` CORE CLOUD FROM V. 2.12.5 You can specify date a timezone jiraDateTime(UTC) Supported timezone arguments	Same as " formatDate " method above, but specifies constant dateformat pattern ("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"), should be compatible with ISO 8601 standard as JIRA requires when setting timestamps to JIRA fields	myfield. jiraDateTime	[e
`escapeJSON` CORE CLOUD	Escapes illegal characters in the field value to generate a valid JSON property. New lines, quotes, tabs and etc will be properly escaped `[entry.myfield.escapeJSON]`	myfield. escapeJSON	[ ]

`asArray`
`asArrayMultiSelect`
`asArrayMultiUserPicker`
CORE
CLOUD
	Tries to create an array from the value. Useful when you want to pass ConfiForms multi-select values to JIRA. Something like
	`"customfield_XXXX" : [[entry.myfield.asArray]]`
	`this will generate`
	`"customfield_XXXX" : ["val1","val2"]`
	`assuming "myfield" field is a multi select and has 2 values: val1 and val2`
	There are variations to support other multi-select fields in JIRA
	https://developer.atlassian.com/jiradev/jira-apis/about-the-jira-rest-apis/jira-rest-api-tutorials/jira-rest-api-example-create-issue
	• asArrayMultiSelect
	• asArrayMultiUserPicker (can be used to generate arrays for both: multi-user and multi-group field types)
	Let us know if something you want to use is missing
		myfield.asArray
		myfield. asArrayMultiSelect
		myfield. asArrayMultiUserPicker
			[
			[
			a
			a

asArray(separator)
FROM V. 1.35
CLOUD
	Same as "asArray" without a parameter, but allows you to set own separator
	Example:
	`asArray(') will wrap the values into list of 'v1', 'v2'...`
	`asArray will do the same with default separator ", like this "v1", "v2"`
	`asArray(_) will output _v1_, _v2_`
		myfield.asArray (separator)	[
			(

asArrayOfIds CORE CLOUD	Same as the above, but exporting IDs of the multi-select values in the following format: `"1", "2"`	myfield.asArrayOfIds	[ ]

asArrayOfKVPairs (key)
FROM V. 1.35
CLOUD
	Will output the list of values in a format:
	`{"key":"value"}, {"key": "value2"}`
	Useful for adding JIRA labels, like when giving a key as "add" asArrayOfKVPairs(add)
	`{"add": "value"}, {"add", "value2"}`
			[
			(
			s
			v

`replaceCRLFWithBR` CORE CLOUD	Replaces CR/LF with **<br/>** tag to show with line brakes in HTML (useful when you reference the textarea field using [entry.] notation)	myfield. replaceCRLFWithBR	[r
`replaceBRWithCRLF` CORE CLOUD	Does the opposite to "replaceCRLFWithBR" and replaces <br/> tags with CRLF		
`asUserFullNames` CORE CLOUD	Works **only with User multi-select fields** and shows list of full names for selected users	myfield. asUserFullNames	[a
`asUserEmails` CORE CLOUD	Works **only with User multi-select fields** and shows list of emails for selected users	myfield. asUserEmails	[ ]
`asUserNames` CORE CLOUD	Works **only with User multi-select fields** and shows list of usernames for selected users	myfield. asUserNames	[a
`friendlyDate` CORE	Formatting date and date/time field types with https://docs.atlassian.com/confluence/latest/com/atlassian/confluence/core/datetime/FriendlyDateFormatter.html	myfield. friendlyDate	[ ]
`asCount` CORE CLOUD	Returns size of a collection for multi-value fields or number of chars for other types	myfield.asCount	[

`asSize` CORE CLOUD	Returns size of a collection for multi-value fields or number of chars for other types (same as "count")	myfield.asSize	[
`asLength` CORE CLOUD	Returns size of a collection for multi-value fields or number of chars for other types	myfield. asLength	[
`formatCurrency` CORE CLOUD	Tries to format value as currency using either default or given format https://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html If value could not be formatted according to given format then value will be returned as is ⓘ Can be used with any number (and not only currency)	myfield. formatCurrency (JAVA_FORMAT) myfield. formatCurrency () (default decimal format is used if empty)	[f ((J e u [f [f
`formatNumber` CORE CLOUD	`alias to formatCurrency`	myfield. formatNumber (JAVA_FORMAT) myfield. formatNumber() (default decimal format is used if empty)	[e [e [e
`asFilteredBy (FILTER)` CORE CLOUD	Very powerful function to extract the value by given filter (mostly used for multi-value fields). Especially useful with Multi-select fields which are of type "smart fields", the ones referencing other forms and fields Example: You have one form which has a field called "mf" which holds multi-value references to another form which has the following fields: name, surname, position Then... when showing the data from the first form you can actually show only specific choices. For example: • entry.mf.asFilteredBy(name:Alex) - to show only persons selected with name Alex only (here "name" references a field in 2nd form) • entry.mf.asFilteredBy(surname:Ve*) - to show only persons selected with surnames starting with "Ve" only (here "surname" references a field in 2nd form) • entry.mf.asFilteredBy(position:CEO) - to filter OUT all those selected who are NOT in CEO position	entry.mf. asFilteredBy (FILTER)	F f p [(T s
`trimAllSpaces` CORE CLOUD	Removes all the spaces in the field value. Can be used when creating page labels automatically from ConfiForms field values and want to ensure the value is taken as a label and not split by spaces into differenet labels	entry.myfield.trimAllSpaces	[s

camelCase CORE CLOUD	Makes a CamelCase string from a given value	entry.myfield. camelCase	[
camelCaseAndTrim CORE CLOUD	Makes a CamelCase string from a given value and, additionally, removes all the spaces	entry.myfield. camelCaseAndTrim	[
addCRLF CORE CLOUD	Adds CR and LF characters after the value	entry.addCRLF	[
addCRLFHtml CORE CLOUD	Adds <br/> (brake) after the value in HTML format	entry. addCRLFHtml	[
trunc(NUMBER) CORE CLOUD truncLeft(NUMBER) CORE CLOUD truncRight(NUMBER) CORE CLOUD	Truncates the value. Leaves "n" first symbols	entry.trunc(100)	[
trim(NUMBER) trim() - will trim the value from leading /trailing spaces CORE CLOUD trimLeft(NUMBER) CORE CLOUD	Trims the value, Skips "n" first symbols	entry.trim(10)	[
trimRight(NUMBER) CORE CLOUD	Trims the value, Removes "n" last symbols		

asAttachment asAttachment(n) CORE	You can reference a particular attachment stored in Confluence and linked though ConfiForms Field (either File or Attachment picker) n - is the index of the attachment stored/linked using ConfiForms Field. Index starts with 0. When no index is specified, then the 1st attachment is taken (1st attachment is stored with index 0) This means that `asAttachment = asAttachment(0)` This function is a "bridge" to get other properties of the attachment stored. See below.	entry. asAttachment	[[[
asAttachment.base64 asAttachment(1). base64 CORE	Returns bas64 encoded string of the file contents in this attachment ConfiForms uses https://commons.apache.org/proper/commons-codec/apidocs/org/apache/commons/codec/binary/Base64.html#encodeBase64URLSafeString-byte:A- to do this and it is important to note that ⓘ Encodes binary data using a URL-safe variation of the base64 algorithm but does not chunk the output. The url-safe variation emits - and _ instead of + and / characters. **Note: no padding is added.** FROM V. 2.24.7 base64 function supports true/false as parameters to generate base64 with or without padding base64(true) - generates URL-safe output (same as without parameter) base64(false) - generates pure base64 hash	entry. asAttachment. base64	[ ] [є [є

asAttachment.
ANY_PROPERTY
asAttachment(n).
ANY_PROPERTY
CORE
	Where ANY_PROPERTY is the "get" methods of the class Attachment https://docs.atlassian.com/confluence/5.9.1/com/atlassian/confluence/pages/Attachment.html
	Examples:
	`asAttachment.displayTitle`
	`asAttachment.id`
	`asAttachment.downloadPath`
	`asAttachment.contentType`
	`and many other properties of the Attachment object`
	`asAttachment(1).displayTitle - also perfectly valid and will try to get the display title for the attachment stored "second" in the field`		

asUsers FROM V. 1.39.2	Converts multi-user / multi-owner (ownedBy field in ConfiForms) field values to list of user objects, which can be then transformed into the desired output as needed, for example: `ownedBy.asUsers.transform(email)` Any property of the User object (see below) is accessible		
asUser.username asUser.fullName asUser.email asUser(n).username asUser(n).fullName asUser(n).email CORE	When working with multi-select user control/field and want to get a particular user info ❗ Only works with Multi-select user field!		

asEntryRef (REF_TO_ENTRY)
CORE
CLOUD
	Example:
	`[entry.id.asEntryRef(entry.id)]`
	This will return
	`[entry.id]`
	When this is necessary?
	The use case is when you use it within an IFTTT or have a ListView which has another ListView inside and you want to prevent ConfiForms from applying the context variables onto the sub-lists
	The function can be used with any field type (as long as the field exists) and actually the following constructions are perfectly valid
	`[entry.id.asEntryRef(entry.somefield)] will return [entry.somefield]`
	`[entry.id.asEntryRef(entry.mytextfield)] will return [entry.mytextfield]`
	`[entry.id.asEntryRef(entry.anotherfield)] will return [entry.anotherfield]`
	`[entry.id.asEntryRef(entry.id)] will return [entry.id]`
	As you can see we apply the function on the same ID field (exists in every ConfiForms record) and the parameter you give in is the important bit in all this...
	So, the parameter (PARAM) you give to **asEntryRef** is wrapped into the brackets and returned:
	`[PARAM]`
	You can also use "_func" pseudo property of a record as a bridge to "asEntryRef" function when the "id" is not available (assigned)
	This is always available
	`[entry._func.asEntryRef(entry.somefield)] will return [entry.somefield]`
	`[entry._func.asEntryRef(entry.mytextfield)] will return [entry.mytextfield]`
	`[entry._func.asEntryRef(entry.anotherfield)] will return [entry.anotherfield]`
	`[entry._func.asEntryRef(entry.id)] will return [entry.id]`		

asVelocityExpRef (VALUE) FROM V. 1.50.1 CLOUD	Same as asEntryRef, and asIFTTTRef, but returns a value wrapped in ${}, like ${VALUE}		
asJSON CORE CLOUD	Converts a value to JSON and allows to access JSON object properties	entry.myfield.asJSON.someJSONProperty entry.myfield.asJSON.anotherJSONProperty	

asRef(VALUE)
FROM V. 1.53.8
CLOUD
	Same as asEntryRef, and asIFTTTRef, but returns the exact experssion as you have put inside the arguments
	asRef(VALUE) will return VALUE		

`asUserProfile` CORE	Converts to User profile (or tries to, if a given field value can be resulved as a user object) Available user profile properties (all standard ones, available in the profile) to reference are: ▪ phone ▪ im ▪ website ▪ position ▪ department ▪ location	entry.somefield.asUserProfile.phone entry.somefield.asUserProfile.im entry.somefield.asUserProfile.website entry.somefield.asUserProfile.position entry.somefield.asUserProfile.department entry.somefield.asUserProfile.location ⓘ where "somefield" should get resolved into username (can be user field)	[a [a [a [a [a [a
asList CORE CLOUD	Extracts values of a list and converts to a comma-separated string, see below for examples on **transform** and **asList**		
asAttachments CORE	Converts the file/attachment field values into the array of Attachment objects https://docs.atlassian.com/confluence/5.9.1/com/atlassian/confluence/pages/Attachment.html		

transform (property_name)

CORE

CLOUD

Converts the list of some objects into the list of values for the given property

For example:

```
[entry.myfile.asAttachments.transform(id)]
```

When "myfile" is a field of type file/attachment, a function "asAttachments" will convert it's values into the list of Attachments objects and then function "transform" will extract and "id" property of each Attachment object and will put it into the resulting list

```
[entry.myfile.asAttachments.transform(id).asList]
```

Same as in previous example, but we convert the otput into a comma separated list of attachment IDs

Below, is another example, which returns a result of ID's, but wrapped into quotes

```
[entry.myfile.asAttachments.transform(id).asArray]
```

Function "transform" can be used on a multi-value field.(on Multi-select fields, which hold the structure of ID and LABEL (properties "id" and "label")) See Accessing field values and properties and dropdown fields

For example, to get the list of **dropdown** IDs

```
[entry.mymultifield.transform(id)]
```

To get the list of **dropdown** labels

```
[entry.mymultifield.transform(label)]
```

To get the list of User's full names for multi-user field (User field has properties: fullName, username, email, lastName, firstname) See Accessing field values and properties

```
[entry.mymultiuserfield.transform(fullName)]
```

same as above, but in "coma-separated" string

```
[entry.mymultiuserfield.transform(fullName).asList]
```

Important!

Transform function also works with smart fields, and could transform the fields / values referenced from another form through the reference (a field) in your current form

See an example here: Using transform function with smart fields

add(number)
CORE
CLOUD
	Adds a numeric value to a field (can substract as well, if a negative number is given)
	`[entry.somedatefield.timestamp.add(86400000)]`
	to add 1 day to the date (value of 86400000 is in milliseconds)
	`[entry.somedatefield.timestamp.add(-86400000)]`
	to add -1 day (substract a day) to the date (value of 86400000 is in milliseconds)
	ⓘ Supports dynamic parameters via [entry.field_name]		

subtract(number) CORE CLOUD	Subtracts value from field value ⓘ Supports dynamic parameters via [entry.field_name]		
multiply(number) CORE CLOUD	Multiplies value by given parameter value `[entry.somedatefield.multiply(2)]` ⓘ Supports dynamic parameters via [entry.field_name]		
divide(number) CORE CLOUD	Divides value by given parameter value `[entry.somedatefield.divide(2)]` ⓘ Supports dynamic parameters via [entry.field_name]		

split(separator)
ⓘ separator could be a space, like:
`split( )`
CLOUD
	Split is an intermediate function to help you with transforming the string values into the arrays of strings, which then can be used with "array type" virtual functions
	For example:
	We have a field type called "mytextfield" and want it's content to be passed to JIRA as labels. For this we need to make sure we split the entered text by "space" and then use a corresponding virtual function to transform the object inti the desired representation
	`mytextfield.split( ).asArrayMultiSelect`
	Any other "array type" function can be applied. For example to get the count, one will write the following
	`mytextfield.split( ).asCount`		

	Now you can easily understand if the field value has been changed or not. Can be used ~~ONLY~~ (from ConfiForms version 1.50.1 can be used in field definition rules as well!) in IFTTT macro, in "condition" parameter. Only in this case, we have a previous snapshot of the data for this record. And the function is applied on the whole entry and not on the field. Example in IFTTT macro:		

hasChanged (fieldName)

FROM V. 1.36

FROM V. 1.50.1

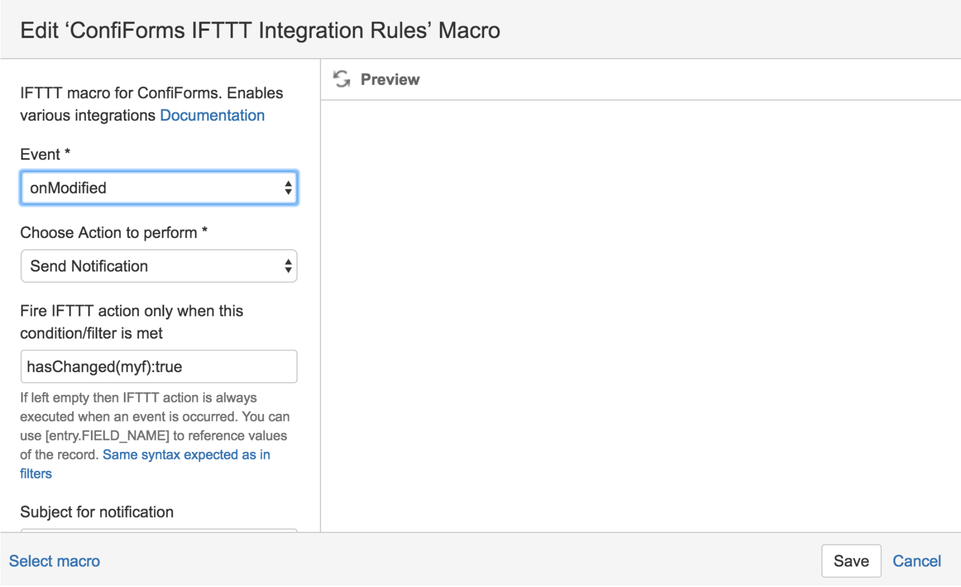
- can be used in Field Definition Rules as well!

FROM V. 1.51.6

- you can use it **without** specifying property to track ANY field change.

By using

```
hasChanged()
:true
```



Edit ‘ConfiForms IFTTT Integration Rules’ Macro

IFTTT macro for ConfiForms. Enables various integrations Documentation

Event *

onModified

Choose Action to perform *

Send Notification

Fire IFTTT action only when this condition/filter is met

hasChanged(myf):true

If left empty then IFTTT action is always executed when an event is occurred. You can use [entry.FIELD_NAME] to reference values of the record. Same syntax expected as in filters

Subject for notification

The result of the function is a boolean, "true" is returned when the value for the field is different from current, and "false" is returned otherwise

Example:

```
hasChanged(somefield):true
```

or

```
hasChanged(somefield):true AND hasChanged(anotherfield):false
```

In addition to "hasChanged" function, ConfiForms has a support to get "previous state" of the record. And that is using a "virtual property" called: "_previousState"

Below example has the same result as "hasChanged(somefield):true"

```
!somefield:[entry._previousState.somefield]
```

This expression could be also used with IFTTT macro condition to determine if the value has been changed, but also allows you to create a more sophisticated filters like the one below:

```
hasChanged(mynum):true AND mynum:<[entry._previousState.mynum]
```

Checks if the value has been changed and if the previous value was bigger than current

> This is available only in IFTTT and this also means that the synthetic property _previousState is available for you to use in filters
>
> For example - we want to run another IFTTT when the record status has changed from one value to another
>
> (when "MyStatus" status field has changed it's value and the value was changed from "requested" to "development")
>
> ```
> hasChanged(MyStatus):true AND MyStatus:development AND _previousState.
> MyStatus:requested
> ```

asUserLink
asUserLinks (same as above, but works on user multiselect fields)
FROM V. 1.36.3
Also, there is a function which renders it as a macro: useful for usage withing a template or LIstView /CalendarView
asUserLinkMacro
	Generates a macro
	`<ac:link><ri:user ri:userkey="USER_KEY_HERE"></ac:link>`
	and renders the HTML out of it
	`USER_KEY_HERE is looked up by function based on the username given`
		entry.myfield. asUserLink	[
			c
			u
			[
			]

asLink
FROM V. 1.37.1
CLOUD
asLinks
FROM V. 1.38
CLOUD
asLink(Some label)
asLink([entry. somefield])
FROM V. 1.44
CLOUD
	Returns value as HTML link (<a href="VALUE">VALUE</a>)
	Also, for collections: function "asLinks"
		entry.myfield. asLink
		entry.myfield. asLink(I am a link label)
			[
			[
			a

acLink CORE acLinkMacro CORE CLOUD	Renders a link to local resource acLinkMacro functions generates a macro only (without rendering to HTML) `<ac:link><ri:page ri:content-title="SPACEKEy:PAGE_TITLE"/></ac:link>`		
asImageLink CORE CLOUD	Renders <img /> link to given url or attachment Id (in cloud)	myfield. asImageLink	[

parseDate(FORMAT)
FROM V. 1.36.6
CLOUD
	Tries to format date fields in the specified format, expects date format as in http://docs.oracle.com/javase/7/docs/api/java/text /SimpleDateFormat.html
	Example on how to parse and format a date from JIRA issue
	`jirakey.fields.created.parseDate(yyyy-MM-dd'T'HH:mm:ss.SSS'Z').formatDate(yyyy-MM-dd)`
	In this example we have a field called "jirakey" in ConfiForms Form, and access a property "created" from JIRA issue.
	Then we parse the created date, using the format JIRA uses when returning the date/time field via REST API and then we format the date using another virtual function called "formatDate" to show it as year-month-day only
		entry.myfield. parseDate (JAVA_FORMAT)	[
			(
			F
			v
			t
			f
			f

toPersianDate CORE	Converts timestamp to Iranian/Persian date `date.toPersianDate`	date. toPersianDate	[

toPersianDateTime CORE	Converts timestamp to Iranian/Persian date with time `mydatefield.toPersianDateTime`	date.toPersianDateTime	[

join
FROM V. 1.38
CLOUD
	You can join the field values into one string. Works best on multi-select fields
	For example (expect a field to be a multi-select field here):
	`id:[entry.mymultifield.transform(id).join( OR id:)]`
	Will extract the ID's from a record stored, and then join (concatenate) those ID's into one string, separated by OR id:
	Something like this (when mymultifield contains 2 items):
	`id:1234-5678-9000-1234-5678-9000 OR id:1234-5678-9000-1234-5678-9001`
		entry.myfield..join(SOME VALUE)	[
			)

asJIRAIssue FROM V. 1.40 CLOUD	Helps you to transform the values of a "JIRA Issues multi-select" field into a JIRA objects to allow access to any property /field of the JIRA issue	entry.myfield.asJIRAIssue.key entry.myfield.asJIRAIssue.fields.customfield_x	A is v J
asInsightObject FROM V. 1.49.1	Helps you to transform the values of a "Insight Objects multi-select" field into a list of Insight Objects, which can be then transformed via "transform" function or properties can be accesssed directy		
asPage FROM V. 1.40 CLOUD	Helps you to transform the values saved to Page objects when used on Page/BlogPosts multi-select field Or to try to transform numeric value to page object (load page by id)	entry.myfield.asPage.spaceKey	A c fi

userInSecurityGroup
securityGroupHas
FROM V. 1.40
CLOUD
	Both functions allow you to check if a particular user belongs to a given security group or not
	Examples:
	If field "u" is holding a username and we want to check if this user belongs to "confluence-administrators" group
	`u.userInSecurityGroup(confluence-administrators)`
	If field "sg" holds a name of a security group and we want top check if a username is in this security group
	`sg.securityGroupHas(sash)`
	In the example below we check the same, but for currently logged in user
	`sg.securityGroupHas()`
	Both functions return "true" when condition is matched and "false" otherwise
	(without quotes)
	This means that if you need to put a filter to check the condition, the full expression will look like:
	`u.userInSecurityGroup(confluence-administrators):true`
	or
	`sg.securityGroupHas():true`		

securityGroups FROM V. 2.0 CLOUD	Retrieves user's securityGroups in a comma separated list `userfield.securityGroups` (Assuming userfiled is the field holding the username)		
securityGroupUsers FROM V. 2.27.14 CLOUD on cloud version we return only the first 200 members of a group	Lists users (comma separated list of user names) for the given security group `fieldContainingSecurotyGroupName.securityGroupUsers`		

get(index)
FROM V. 1.42.4
CLOUD
Using get() without an index will return the last item from the list
FROM V. 1.44.2
CLOUD
	Get's the element by index form a multi-value field
	To get the first element (index starts from 0)
	`somefield.get(0)`
	Another example with chaining the functions (when somefield holds a string value and we split it by "," and get the first element)
	`somefield.split(,).get(0)`		

append
FROM V. 1.43
CLOUD
	Appends text to a value
	Will append a space to the value of "somefield"
	`somefield.append( )`
	For multi-select dropdown, adding space after transformation and shows in as html link
	`somefield.transform(label.append()).asLinks`
	ⓘ Argument can be dynamic, and reference another field, like [entry.somefield], so you can concat values together	

prepend FROM V. 1.45.2 CLOUD	Prepends text to a value Will append a space to the value of "somefield" `somefield.prepend(SOME TEXT HERE)` ⓘ Argument can be dynamic, and reference another field, like [entry.somefield], so you can concat values together	
lowerCase FROM V. 1.44.2 CLOUD	Returns a lowercases value for the field	
upperCase FROM V. 1.44.2 CLOUD	Returns a uppercased value for the field	
formatLinks FROM V. 1.45.2 CLOUD	Tries to format the http(s) links found in the text as HTML links `somefield.formatLinks`	

greenhopperAsJSON
FROM V. 1.45.3
CLOUD
	Tries to parse the fiedl value returned by JIRA API for greenhopper fields
	`com.atlassian.greenhopper.service.sprint.Sprint@71f1f2ae[id=6745,`
	`rapidViewId=2391,state=ACTIVE,name=My Sprint 2,startDate=2017-08-29T10:46:33.923`
	`+01:00,endDate=2017-09-08T10:46:00.000+01:00,completeDate=<null>,sequence=6745]`
	to a structure you can access via properties
	`id=6745`
	`rapidViewId=2391`
	`state=ACTIVE`
	`name=My Sprint 2`
	`startDate=2017-08-29T10:46:33.923+01:00`
	`endDate=2017-09-08T10:46:00.000+01:00`
	`completeDate=`
	`sequence=6745`
	Example (to return a sprint name)
	`myJIRAField.fields.customfield_10900.greenhopperAsJSON.name`		

asIFTTTRef (RESULT_NAME_AND_MORE)
FROM V. 1.47.2
CLOUD
	The idea and the need for this function is the same as described for function *asEntryRef(REF_TO_ENTRY)*
	This allows you to "escape" the ${IftttResult_NAME.someproperty} into a function to workaround template evaluations against the current record. Useful when your ConfiForms Form creates a page with another ConfiFormiForms Form and that form has various rules using iftttResults or entry.fieldnames
	`entry.id.asIFTTTRef(MYRESULT.id) will produce ${iftttResult_MYRESULT.id}`
	`entry.id.asIFTTTRef(MYRESULT) will produce ${iftttResult_MYRESULT}`		

asUserProfileLink FROM V. 1.48	Outputs a field value as a link to user profile (with avatar and full name). The field must have a username as a value `[entry.myfieldholdingusername.asUserProfileLink]`		

<table>
<tr><td>replaceAccents
replaceAccents()
FROM V. 1.48.2
CLOUD</td><td>Replaces the following accents letters using the folllwing mapping
"Ä" -> "Ae"<br>"Æ" -> "Ae"<br>"ä" -> "ae"<br>"æ" -> "ae"<br>"Ö" -> "Oe"<br>"ö" -> "oe"<br>"Ü" -> "Ue"<br>"ü" -> "ue"<br>"ß" -> "ss"<br>"ó" -> "o"<br>"ú" -> "u"<br>"Ç" -> "C"<br>"ç" -> "c"<br>"í" -> "i"<br>"Ñ" -> "N"<br>"ñ" -> "n"<br>"À" -> "A"<br>"Â" -> "A"<br>"à" -> "a"<br>"â" -> "a"<br>"È" -> "E"<br>"É" -> "E"<br>"Ê" -> "E"<br>"è" -> "e"<br>"é" -> "e"<br>"ê" -> "e"
[entry.myfield.replaceAccents]</td><td></td><td></td></tr>
<tr><td>replaceWith (searchstring, replace with)
FROM V. 1.48.2
CLOUD</td><td>Replaces every value matching "search string" in a field with given "replace with" value
[entry.myfield.replaceWith(hi,hola)]
Will match every "hi" and replace it with "hola"</td><td></td><td></td></tr>
<tr><td>formatLink(url)
formatLink(url|label)
FROM V. 1.51.3
CLOUD</td><td>Helps you to create http links from values in ConfiForms directly in teh views
[entry.myfield.formatLink(https://google.com?q=)]
will create a <a href="https://google.com?q=<VALUE_OF_MY_FIELD>" target="_blank" ><VALUE_OF_MY_FIELD></a>
the below example shows how to specify a contant label for your links
[entry.myfield.formatLink(https://google.com?q=|search)]
will create a <a href="https://google.com?q=<VALUE_OF_MY_FIELD>" target="_blank" >search</a></td><td></td><td></td></tr>
<tr><td>timestamp
FROM V. 1.52.1
CLOUD</td><td>Tries to get the timestamp from date/datetime/timestamp holding fields in epoch format</td><td></td><td></td></tr>
</table>

<table>
<tr>
<td>asHex
FROM V. 1.52.1
CLOUD</td>
<td>Converts string into hex representation
<table>
<tr><th>string</th><th>asHex</th></tr>
<tr><td>1</td><td>31</td></tr>
<tr><td>2</td><td>32</td></tr>
<tr><td>hello</td><td>68656c6c6f</td></tr>
</table>
You can use it with other functions, as usual... something like
<pre>
[entry.myfield.asHex.upperCase]
</pre>
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dec2Hex
FROM V. 1.52.1
CLOUD</td>
<td>Converts decimals to hex
<table>
<tr><th>value</th><th>dec2Hex</th></tr>
<tr><td>1</td><td>00000001</td></tr>
<tr><td>2</td><td>00000002</td></tr>
<tr><td>255</td><td>000000FF</td></tr>
<tr><td>210</td><td>000000D2</td></tr>
</table>
As always, you can chain the functions
<pre>
[entry.myfield.dec2Hex.upperCase]
[entry.myfield.dec2Hex.trim(4)] - returns last 4 digits, instead of 000000D2 for
210 will return 00D2
</pre>
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>randomInt(minVal)
FROM V. 2.0.8</td>
<td>You can generate tips from ConfiForms data with something like
<pre>
<ac:macro ac:name="confiform-plain">
  <ac:parameter ac:name="filter">valuecounter:[entry._total.randomInt(1)]</ac:
parameter>
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
  <ac:plain-text-body><![CDATA[[entry.advice]]]></ac:plain-text-body>
</ac:macro>
</pre>
To show "random" advice from your form of "advices"

Where "valuecounter" field is autonumber field and we randomly picking one record from a dataset, starting from 1 (that is why we use "randomInt(1)" function on _total field)
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>toString
CORE</td>
<td>Ensures the value is a string value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timezoneAwareDate
CORE

Since
2.13.7
supports setting the formatting pattern

timezoneAwareDate(PATTERN)
CLOUD</td>
<td>Tries to format the date given in the user's timezone. Uses user's defined formatting pattern to format the date.

Unless given as parameter (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html)

Date is always in the user's timezone</td>
<td></td>
<td></td>
</tr>
</table>

<table>
<tr><td>timezoneAwareDateTime
CORE
Since
2.13.7
supports setting the formatting pattern
timezoneAwareDateTime(PATTERN)
CLOUD</td><td>Tries to format the date/time given in the user's timezone. Uses user's defined formatting pattern to format the datetime.
Unless given as parameter (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html)
Date is always in the user's timezone</td><td></td><td></td></tr>
<tr><td>formatNumberWithLocale(COUNTRY)
CORE</td><td>Formats number with a given locale
formatNumberWithLocale(de) will format the number in German locale.
See supported locales in Java 8, https://www.oracle.com/technetwork/java/javase/java8locales-2095355.html</td><td></td><td></td></tr>
<tr><td>compactDateTimeInterval
FROM V. 2.0.23</td><td>Helper function to render DateTime Interval field values in a more compact way when the date for start and end is within the same day
[entry.mydatetimeintervalfield.compactDateTimeInterval]
Example (the output format is specific to your Confluence date/time formatting settings):
<table>
<tr><th>DateTime Interval value</th><th>Output</th><th></th></tr>
<tr><td>Mar 19, 2019 2:00 PM -
Mar 19, 2019 3:00 PM</td><td>Mar 19, 2019 2:00 PM – 3:
00 PM</td><td>Transforms to a short format</td></tr>
<tr><td>Mar 19, 2019 2:00 PM -
Mar 20, 2019 3:00 PM</td><td>Mar 19, 2019 2:00 PM -
Mar 20, 2019 3:00 PM</td><td>Stays in long format, as event spans 2 days</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>truncWithExpand(N) where N is the number of characters to show
FROM V. 2.0.25
CLOUD</td><td>[entry.field_name.truncWithExpand(10)]
will show the first 10 symbols and if the value is longer then the "..." block will be shown to allow your users to expand the value</td><td></td><td></td></tr>
<tr><td>extractText
CORE
CLOUD</td><td>Extracts text from HTML value</td><td></td><td></td></tr>
<tr><td>renderAsText
CORE
CLOUD</td><td>Render as Text (mainly to be used with values produced by wiki markdown field type)</td><td></td><td></td></tr>
<tr><td>renderAsHtml
CORE
CLOUD</td><td>Render as HTML (mainly to be used with values produced by wiki markdown field type)</td><td></td><td></td></tr>
</table>

renderWikiMarkup FROM V. 2.1.0	Renders wiki markup contents as html		
removeCRLFs FROM V. 2.1.0 CLOUD	Removes all new lines in a field value		
removeSpaces FROM V. 2.1.0 CLOUD	Removes all the spaces in the field value		
remove(value) CORE CLOUD	Removes value from the field (works as replaceWith for non-collections (single fields) and as remove item for multi-value fields) If you give a CSV values for multi-select field to *remove* then it will attempt to remove each value (splitting by ,)		
obscure FROM V. 2.6.0 CLOUD	Hides the value behind the given mask `pwd.obscure(********|click to view)` Will create something like ********(click to view) Clicking on the "click to view" link will reveal the actual value of the "pwd" field		

hasValue(value)
FROM V. 2.9.4
CLOUD
	Helps you to determine if the field has certain value. Useful when you want to check if the multi-select field has a certain option selected
	`field.hasValue(somevalue)`
	For single choice fields it checks for equality
	Can be used with Supported math operators, formulas and functions to construct conditional formulas
	For example:
	`IF([entry.Options.hasValue(choice1)], 10, 0)`
	Or in filtering conditions, checking for true or false:
	`Options.hasValue(choice1):true`
	Of course, chaining with *transform* is absolutely possible
	`Options.transform(label).hasValue(my label):true`
	or (assuming Options is a multi-user field)
	`Options.transform(fullName).hasValue(John Smith):true`		

escapeSQL FROM V. 2.9.5 CLOUD	Escapes SQL parameters		

getOptions *FROM V. 2.9.5* *CLOUD*	Allows you to get all the options registered with a field (choice based field, like radio group, checkbox group, dropdowns...) For example, to get the labels for options registered in a field called "radiogroup" `id.getOptions(radiogroup).transform(label)`		
getUnselectedOptions *FROM V. 2.9.5* *CLOUD*	Allows you to get all the UNSELECTED options registered with a field (choice based field, like radio group, checkbox group, dropdowns... To get unselected options from a field called "radiogroup" and show them on each line separately `id.getUnselectedOptions(radiogroup).transform(label.append(<br/>))`		

queryAndRender (<formName: pageId>;<FILTER>;<FIELDS_OR_EXPRESSIONS>;<VIEW_TYPE>)
FROM V. 2.10
CLOUD
	You can add a ConfiForms Field to your form's view and set it to show the contents of another form within your form
	- <formName:pageId> - name of the form and page id (location of the form) separated by :
	- <FILTER> - ConfiForms Filters, can be dynamic and reference values in a current record via [entry.field_name]
	- <FIELDS_OR_EXPRESSIONS> - list of fields (can contain expressions or virtual functions) to render from the referenced form separated by |
	- <VIEW_TYPE> - how the view shall be rendered. Supported values at this point are:
	○ table
	○ card
	○ list
	Example
	`id.queryAndRender(f:819201;*;t;table)`
	this renders a TableView inside the field for form "f" located on page "819201", showing all the records from this form and only "t" field.
	`id.queryAndRender(myform:819202;partner:[entry.partner];name|amount;card)`
	this renders a CardView inside the field for form "myform" located on page "819202", showing records matching the value in "partner" field and showing 2 fields: "name" and "amount".		

queryAndSet(<formName:pageId>;<FILTER>;<FIELD_OR_EXPRESSION>)
FROM V. 2.16.9
CLOUD
Support for Sorting in ConfiForms parameter as shown below
FROM V. 2.21.3
queryAndSet(<formName:pageId>;<FILTER>;<FIELD_OR_EXPRESSION>;name AS LIMIT 1)
	You can set up this function to lookup (query) the value from some form by given filter and you can set the result of this query to a field
	Useful when you want to make a value lookup in ConfiForms IFTTT
	- <formName:pageId> - name of the form and page id (location of the form) separated by :
	- <FILTER> - ConfiForms Filters, can be dynamic and reference values in a current record via [entry.field_name]
	- <FIELD_OR_EXPRESSION> - field to get the value form.
	When multiple records much the filter the value will be concatenated
	Example:
	`id.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield)`
	same can be done though the _func helper
	`_func.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield)`
	`or`
	`[entry._func.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield)]`
	FROM V. 2.21.3
	`id.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield;LIMIT 1)`
	`id.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield;created DESC LIMIT 1)`
	FROM V. 2.25.2
	`id.queryAndSet(myform:1111222;myfield:[entry.field];sum:(anotherfield);created DESC LIMIT 1)`
	See **sum:(anotherfield)**, as an expression to give for the field you want to query on - this way it will attempt to "sum up" found values if all the found values are of numeric type
	Similarly you can just warp the field or and expression into brackets
	`(anotherfield)`
	`([entry.anotherfield])`		

pageProperties(property_name)
FROM V. 2.10.7
CLOUD
	Access page properties of a page object. The field you can apply this function has to be a page or should resolve into a page (page ID)
	`mypage.pageProperties(property_name)`
	What is page properties: https://confluence.atlassian.com/doc/page-properties-macro-184550024.html		

sort(property_name ASC)
sort(property_name1 ASC, property_name2 DESC)
FROM V. 2.11.3
CLOUD
	Sorts lists of choices / objects		

limit(number)
FROM V. 2.11.3
CLOUD
	Limits the number of items in an array/list		

asUserTimezone *FROM V. 2.12.4* *CLOUD*	Represents given date as in user timezone date. Actually al the dates in ConfiForms are stored/entered in server timezone! But sometimes it is required to pass the selected date further (to Jira, for example) as if it is given in user timezone.	someDate.asUserTimezone.jiraDateTime	[a f
asUserTimezoneAwareDate asUserTimezoneAwareDateTime *FROM V. 2.12.4* Since **2.13.7** supports setting the formatting pattern asUserTimezoneAwareDate(PATTERN) asUserTimezoneAwareDateTime(PATTERN) *CLOUD*	Aliases for *timezoneAwareDate* and *timezoneAwareDateTime*	Formats dates in user timezones someDate.asUserTimezoneAwareDate someDate.asUserTimezoneAwareDateTime	
timezoneOffset *FROM V. 2.13.2* *CLOUD*	Returns offset between server and user's timezones in milliseconds		
htmlToWiki *FROM V. 2.13.2* *CLOUD*	Attempts to convert an HTML to Atlassian wiki markup (please note that this is an experimental function and does not support nested inline CSS styles)		

iterateAndPrint
FROM V. 2.13.10
	Now it is possible to iterate through the multi-select fields and print their properties in a convenient way
	For example, iterating on multi user field and printing username and full name of the user in a row for each selected user
	`mymultiuserfield.iterateAndPrint([entry.username] - [entry.fullName] <br/>).renderAsHtml`
	For comment field
	`mycomment.iterateAndPrint(<p>[entry.username] - [entry.timestamp.formatDate]:[entry.content]</p>).renderAsHtml`		

formatToPattern *FROM V. 2.13.10* *CLOUD*	Formats value to given regular expression pattern. For example, to format the phone number given 1234567890 into (123) 456-7890 You will need to do something like this:\\ `myfield.formatToPattern((\d{3})(\d{3})(\d+), ($1) $2-$3)` Where first parameter is a grouping regular expression: ***(\d{3})(\d{3})(\d+)*** And the second one is how the "groups" should be printed (pattern itself): ***($1) $2-$3***		
storageToHtml *FROM V. 2.14* *CLOUD*	Attempts to convert/render given value into HTML. Will work when the value is a valid storage format (Atlassian page storage format)		

evaluateFormula (FORMULA)
FROM V. 2.17
CLOUD
	Evaluates and calculates a given formula. Formula should be given as explained here: Supported math operators, formulas and functions
	Parameters can be given as [entry.field_name] references
	Can be applied on a field or on _func for convenience
	`[entry._func.evaluateFormula(IF(EMPTY("[entry.somefield]"), "ERROR", "ALL GOOD"))]`
	Or in ConfiForms Field macros it is easier to use via "id" field
	`id.evaluateFormula(IF(EMPTY("[entry.somefield]"), "ERROR", "ALL GOOD"))`		

formatDuration (PATTERN)
FROM V. 2.19.4
CLOUD
	This function helps to format numeric value in milliseconds as "duration"
	Example:
	`somefield.formatDuration(H:mm:ss)`
	`somefield.formatDuration(d H:mm:ss)`
	`somefield.formatDuration(d 'days' H:mm:ss)`
	Value of "somefield" needs to be numeric and represent time in milliseconds		

<table>
<tr>
<td>round(scale, roundingMode)

FROM V. 2.20.2

CLOUD</td>
<td>Where scale must be an integer

And <em>rounding mode</em> could be given as a numeric value or as text value

<table>
<tr><th>Text constant</th><th>Numeric</th></tr>
<tr><td><code>UP</code></td><td>0</td></tr>
<tr><td><code>DOWN</code></td><td>1</td></tr>
<tr><td><code>CEILING</code></td><td>2</td></tr>
<tr><td><code>FLOOR</code></td><td>3</td></tr>
<tr><td><code>HALF_UP</code></td><td>4</td></tr>
<tr><td><code>HALF_DOWN</code></td><td>5</td></tr>
<tr><td><code>HALF_EVEN</code></td><td>6</td></tr>
<tr><td><code>UNNECESSARY</code></td><td>7</td></tr>
</table>

Example:
<pre>somenumericfield.round(2, HALF_UP)</pre>same as
<pre>somenumericfield.round(2, 4)</pre></td>
<td></td>
<td></td>
</tr>
<tr>
<td>barcode(type)

FROM V. 2.26

CLOUD</td>
<td>Renders value as barcode

Supported types are:
<ul>
<li>Interleaved2of5</li>
<li>Codabar</li>
<li>ITF-14</li>
<li>Code39</li>
<li>Code128</li>
<li>EAN-128</li>
<li>GS1-128</li>
<li>UPC-A</li>
<li>UPC-E</li>
<li>EAN-13</li>
<li>EAN-8</li>
<li>POSTNET</li>
<li>RoyalMail</li>
<li>USPS</li>
<li>PDF417</li>
</ul>
<pre>somevalue.barcode(Codabar)</pre></td>
<td>somevalue.
barcode
(Codabar)</td>
<td>[
(</td>
</tr>
</table>

qrcode
qrcode(type)
FROM V. 2.26
CLOUD
	Renders values as QR code
	Supported types are:
	• url *(default)*
	• SMS
	• MMS
	• email
	Suggesting to use SMS when scanned (field value is preset as phone number)
	`somevalue.qrcode(SMS)`
	Or as default (url) - field value is set as URL
	`somevalue.qrcode`
		somevalue. qrcode
		somevalue. qrcode(MMS)
			[
			[(

isValidPassword
FROM V. 2.27.24
	`value.isValidPassword()`
	`value.isValidPassword`
	Returns true or false
	Checks the password given in the field value for current user only
	In other words this function can only verify current user's password
		value. isValidPassword():true
		value. isValidPassword():false
		value. isValidPassword:true
		value. isValidPassword:false
			[i
			[i

isValidCaptcha
FROM V. 2.27.25
	`field.isValidCaptcha()`
	`field.isValidCaptcha`
	Returns true or false
		field. isValidCaptcha():true
		field. isValidCaptcha():false
		field. isValidCaptcha
		field. isValidCaptcha
			[i
			[

variables FROM V. 2.27.26	You can access the value of a variable defined in ConfiForms configuration: "Variables and Secrets" section. Please note that you can only access variables, but not secrets. **The value for the secret will always be returned as empty value** `variables(variable_name)`	field.variables (variable_name)	

queryCount (<formName: pageId>;<FILTER>)
FROM V. 2.27.29
CLOUD
	You can lookup records number in the form matching your filter
	`id.queryCount(myform:1111222;myfield:[entry.field])`
	You can refermce to the same page via @self (if your form is located on the same page)
	`id.queryCount(myform:@self;*)`
	Returns number of records matching (numeric)		

If entry value is null or could not be formatted according to rules then value is returned as-is

ⓘ See also Accessing field values and properties. You can use complex properties in your filters. For example filtering dropdown fields by values and by labels, filtering page type fields by page metadata fields, filtering user fields by, for example - email property

It is important to understand that it is **totally possible to combine virtual functions WITH field properties and to chain functions!**