

Virtual functions

FROM V. 1.40



See if the function is available for [ConfiForms CLOUD](#) -

CLOUD

In order to make it easier to integrate with other external systems, such as JIRA, for example, 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]**
- When using in **ConfiForms Field macro** then reference it by fieldname, 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



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
<code>urlencode</code> 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
<code>escapeXML</code> 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
<code>escape</code> 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. escapeForHtmlAttribute
<code>escapeJavaScript</code> CORE CLOUD	Escapes JavaScript from the value	


<p>formatDate</p> <p>CORE</p> <p>CLOUD</p> <p>FROM V. 2.12.5</p> <p>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</p> <p>Example:</p> <p>formatDate(MM-dd-yyyy, UTC)</p> <p>Supported timezone arguments</p>	<p>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</p>	<p>myfield.formatDate (JAVA_FORMAT)</p>
<p>convertDate (FORMAT, TIMEZONE)</p> <p>FROM V. 2.13.5</p> <p>Supported timezone arguments</p>	<p>This function converts a timestamp (UTC) into the formatted date in a specified timezone</p> <p>Format can be anything supported by http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html</p> <p>Timezone argument can be anything documented here: Supported timezone arguments</p>	<p>Example:</p> <p>myfield.convertDate(yy-MM-dd'T'HH:mm:ss.SSSZ, Europe/Tallinn)</p>
<p>jiraDate</p> <p>CORE</p> <p>CLOUD</p> <p>FROM V. 2.12.5</p> <p>You can specify date a timezone</p> <p>jiraDate(America/Chicago)</p> <p>Supported timezone arguments</p>	<p>Same as "formatDate" method above, but specifies constant DateFormat pattern (yyyy-MM-dd), suitable for JIRA REST API</p>	<p>myfield.jiraDate</p>
<p>jiraDateTime</p> <p>CORE</p> <p>CLOUD</p> <p>FROM V. 2.12.5</p> <p>You can specify date a timezone</p> <p>jiraDateTime(UTC)</p> <p>Supported timezone arguments</p>	<p>Same as "formatDate" method above, but specifies constant dateFormat pattern ("yyyy-MM-dd'T'HH:mm:ss.SSSZ"), should be compatible with ISO 8601 standard as JIRA requires when setting timestamps to JIRA fields</p>	<p>myfield.jiraDateTime</p>
<p>escapeJSON</p> <p>CORE</p> <p>CLOUD</p>	<p>Escapes illegal characters in the field value to generate a valid JSON property. New lines, quotes, tabs and etc will be properly escaped</p> <pre>[entry.myfield.escapeJSON]</pre>	<p>myfield.escapeJSON</p>

<p>asArray</p> <p>asArrayMultiSelect</p> <p>asArrayMultiUserPicker</p> <p>CORE</p> <p>CLOUD</p>	<p>Tries to create an array from the value. Useful when you want to pass ConfiForms multi-select values to JIRA. Something like</p> <pre>"customfield_XXXX" : [[entry.myfield.asArray]]</pre> <p>this will generate</p> <pre>"customfield_XXXX" : ["val1","val2"]</pre> <p>assuming "myfield" field is a multi select and has 2 values: val1 and val2</p> <p>There are variations to support other multi-select fields in JIRA</p> <p>https://developer.atlassian.com/jiradev/jira-apis/about-the-jira-rest-apis/jira-rest-api-tutorials/jira-rest-api-example-create-issue</p> <ul style="list-style-type: none"> asArrayMultiSelect asArrayMultiUserPicker (can be used to generate arrays for both: multi-user and multi-group field types) <p>Let us know if something you want to use is missing</p>	<p>myfield.asArray</p> <p>myfield.asArrayMultiSelect</p> <p>myfield.asArrayMultiUserPicker</p>
<p>asArray(separator)</p> <p>FROM V. 1.35</p> <p>CLOUD</p>	<p>Same as "asArray" without a parameter, but allows you to set own separator</p> <p>Example:</p> <pre>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_</pre>	<p>myfield.asArray(separator)</p>
<p>asArrayOfIds</p> <p>CORE</p> <p>CLOUD</p>	<p>Same as the above, but exporting IDs of the multi-select values in the following format:</p> <pre>"1", "2"</pre>	<p>myfield.asArrayOfIds</p>
<p>asArrayOfKVPairs(key)</p> <p>FROM V. 1.35</p> <p>CLOUD</p>	<p>Will output the list of values in a format:</p> <pre>{"key": "value"}, {"key": "value2"}</pre> <p>Useful for adding JIRA labels, like when giving a key as "add" asArrayOfKVPairs(add)</p> <pre>{"add": "value"}, {"add", "value2"}</pre>	
<p>replaceCRLFwithBR</p> <p>CORE</p> <p>CLOUD</p>	<p>Replaces CR/LF with
 tag to show with line brakes in HTML (useful when you reference the textarea field using [entry.] notation)</p>	<p>myfield.replaceCRLFwithBR</p>
<p>replaceBRwithCRLF</p> <p>CORE</p> <p>CLOUD</p>	<p>Does the opposite to "replaceCRLFwithBR" and replaces
 tags with CRLF</p>	

<p>asUserFullNames</p> <p>CORE</p> <p>CLOUD</p>	<p>Works only with User multi-select fields and shows list of full names for selected users</p>	<p>myfield. asUserFullNames</p>	<p>[a</p>
<p>asUserEmails</p> <p>CORE</p> <p>CLOUD</p>	<p>Works only with User multi-select fields and shows list of emails for selected users</p>	<p>myfield. asUserEmails</p>	<p>[]</p>
<p>asUserNames</p> <p>CORE</p> <p>CLOUD</p>	<p>Works only with User multi-select fields and shows list of usernames for selected users</p>	<p>myfield. asUserNames</p>	<p>[a</p>
<p>friendlyDate</p> <p>CORE</p>	<p>Formatting date and date/time field types with https://docs.atlassian.com/confluence/latest/com/atlassian/confluence/core/datetime/FriendlyDateFormatter.html</p>	<p>myfield. friendlyDate</p>	<p>[]</p>
<p>asCount</p> <p>CORE</p> <p>CLOUD</p>	<p>Returns size of a collection for multi-value fields or number of chars for other types</p>	<p>myfield.asCount</p>	<p>[</p>
<p>asSize</p> <p>CORE</p> <p>CLOUD</p>	<p>Returns size of a collection for multi-value fields or number of chars for other types (same as "count")</p>	<p>myfield.asSize</p>	<p>[</p>
<p>asLength</p> <p>CORE</p> <p>CLOUD</p>	<p>Returns size of a collection for multi-value fields or number of chars for other types</p>	<p>myfield. asLength</p>	<p>[</p>
<p>formatCurrency</p> <p>CORE</p> <p>CLOUD</p>	<p>Tries to format value as currency using either default or given format</p> <p>https://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html</p> <p>If value could not be formatted according to given format then value will be returned as is</p> <div data-bbox="332 1507 1318 1591" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Can be used with any number (and not only currency)</p> </div>	<p>myfield. formatCurrency (JAVA_FORMAT)</p> <p>myfield. formatCurrency ()</p> <p>(default decimal format is used if empty)</p>	<p>[f ((J e t [f [f</p>

<p>formatNumber</p> <p>CORE</p> <p>CLOUD</p>	<p>alias to formatCurrency</p>	<p>myfield. formatNumber (JAVA_FORMAT)</p> <p>myfield. formatNumber()</p> <p>(default decimal format is used if empty)</p>	<p>[e [e [e</p>
<p>asFilteredBy (FILTER)</p> <p>CORE</p> <p>CLOUD</p>	<p>Very powerful function to extract the value by given filter (mostly used for multi-value fields).</p> <p>Especially useful with Multi-select fields which are of type "smart fields", the ones referencing other forms and fields</p> <p>Example:</p> <p>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</p> <p>Then... when showing the data from the first form you can actually show only specific choices.</p> <p>For example:</p> <ul style="list-style-type: none"> 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 	<p>entry.mf. asFilteredBy (FILTER)</p>	<p>F f F [(T s</p>
<p>trimAllSpaces</p> <p>CORE</p> <p>CLOUD</p>	<p>Removes all the spaces in the field value.</p> <p>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 different labels</p>	<p>entry.myfield.trimAllSpaces</p>	<p>[s</p>
<p>camelCase</p> <p>CORE</p> <p>CLOUD</p>	<p>Makes a CamelCase string from a given value</p>	<p>entry.myfield.camelCase</p>	<p>[</p>
<p>camelCaseAndTrim</p> <p>CORE</p> <p>CLOUD</p>	<p>Makes a CamelCase string from a given value and, additionally, removes all the spaces</p>	<p>entry.myfield.camelCaseAndTrim</p>	<p>[A</p>
<p>addCRLF</p> <p>CORE</p> <p>CLOUD</p>	<p>Adds CR and LF characters after the value</p>	<p>entry.addCRLF</p>	<p>[</p>
<p>addCRLFhtml</p> <p>CORE</p> <p>CLOUD</p>	<p>Adds
 (brake) after the value in HTML format</p>	<p>entry.addCRLFhtml</p>	<p>[</p>

<p>trunc(NUMBER)</p> <p>CORE</p> <p>CLOUD</p> <p>truncLeft(NUMBER)</p> <p>CORE</p> <p>CLOUD</p> <p>truncRight(NUMBER)</p> <p>CORE</p> <p>CLOUD</p>	<p>Truncates the value. Leaves "n" first symbols</p>	<p>entry.trunc(100)</p>
<p>trim(NUMBER)</p> <p>trim() - will trim the value from leading /trailing spaces</p> <p>CORE</p> <p>CLOUD</p> <p>trimLeft(NUMBER)</p> <p>CORE</p> <p>CLOUD</p>	<p>Trims the value, Skips "n" first symbols</p>	<p>entry.trim(10)</p>
<p>trimRight(NUMBER)</p> <p>CORE</p> <p>CLOUD</p>	<p>Trims the value, Removes "n" last symbols</p>	
<p>asAttachment</p> <p>asAttachment(n)</p> <p>CORE</p>	<p>You can reference a particular attachment stored in Confluence and linked though ConfiForms Field (either File or Attachment picker)</p> <p>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)</p> <p>This means that</p> <pre>asAttachment = asAttachment(0)</pre> <p>This function is a "bridge" to get other properties of the attachment stored. See below.</p>	<p>entry.asAttachment</p>
<p>asAttachment.base64</p> <p>asAttachment(1).base64</p> <p>CORE</p>	<p>Returns bas64 encoded string of the file contents in this attachment</p>	<p>entry.asAttachment.base64</p>
<p>asAttachment.ANY_PROPERTY</p> <p>asAttachment(n).ANY_PROPERTY</p> <p>CORE</p>	<p>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</p> <p>Examples:</p> <pre>asAttachment.displayTitle asAttachment.id asAttachment.downloadPath asAttachment.contentType</pre> <p>and many other properties of the Attachment object</p> <pre>asAttachment(1).displayTitle - also perfectly valid and will try to get the display title for the attachment stored "second" in the field</pre>	

<p>asUsers</p> <p>FROM V. 1.39.2</p>	<p>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:</p> <pre>ownedBy.asUsers.transform(email)</pre> <p>Any property of the User object (see below) is accessible</p>	
<p>asUser.username asUser.fullName asUser.email asUser(n).username asUser(n).fullName asUser(n).email</p> <p>CORE</p>	<p>When working with multi-select user control/field and want to get a particular user info</p> <div style="border: 1px solid red; padding: 5px; background-color: #ffe6e6;">  Only works with Multi-select user field! </div>	
<p>asEntryRef (REF_TO_ENTRY)</p> <p>CORE</p> <p>CLOUD</p>	<p>Example:</p> <pre>[entry.id.asEntryRef(entry.id)]</pre> <p>This will return</p> <pre>[entry.id]</pre> <p>When this is necessary?</p> <p>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</p> <p>The function can be used with any field type (as long as the field exists) and actually the following constructions are perfectly valid</p> <pre>[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]</pre> <p>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...</p> <p>So, the parameter (PARAM) you give to asEntryRef is wrapped into the brackets and returned:</p> <pre>[PARAM]</pre> <p>You can also use "_func" pseudo property of a record as a bridge to "asEntryRef" function when the "id" is not available (assigned)</p> <p>This is always available</p> <pre>[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]</pre>	

<p>asVelocityExpRef (VALUE)</p> <p>FROM V. 1.50.1</p> <p>CLOUD</p>	<p>Same as asEntryRef, and asIFTTTRef, but returns a value wrapped in \${}, like \${VALUE}</p>	
<p>asJSON</p> <p>CORE</p> <p>CLOUD</p>	<p>Converts a value to JSON and allows to access JSON object properties</p>	<p>entry.myfield.asJSON.someJSONProperty</p> <p>entry.myfield.asJSON.anotherJSONProperty</p>
<p>asRef(VALUE)</p> <p>FROM V. 1.53.8</p> <p>CLOUD</p>	<p>Same as asEntryRef, and asIFTTTRef, but returns the exact expression as you have put inside the arguments</p> <p>asRef(VALUE) will return VALUE</p>	
<p>asUserProfile</p> <p>CORE</p>	<p>Converts to User profile (or tries to, if a given field value can be resolved as a user object)</p> <p>Available user profile properties (all standard ones, available in the profile) to reference are:</p> <ul style="list-style-type: none"> ▪ phone ▪ im ▪ website ▪ position ▪ department ▪ location 	<p>entry.somefield.asUserProfile.phone</p> <p>entry.somefield.asUserProfile.im</p> <p>entry.somefield.asUserProfile.website</p> <p>entry.somefield.asUserProfile.position</p> <p>entry.somefield.asUserProfile.department</p> <p>entry.somefield.asUserProfile.location</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i where "so mefield" should get resolved into use name (can be user field)</p> </div>
<p>asList</p> <p>CORE</p> <p>CLOUD</p>	<p>Extracts values of a list and converts to a comma-separated string, see below for examples on transform and asList</p>	
<p>asAttachments</p> <p>CORE</p>	<p>Converts the file/attachment field values into the array of Attachment objects</p> <p>https://docs.atlassian.com/confluence/5.9.1/com/atlassian/confluence/pages/Attachment.html</p>	

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 output 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](#)

<p>add(number)</p> <p>CORE</p> <p>CLOUD</p>	<p>Adds a numeric value to a field (can subtract as well, if a negative number is given)</p> <pre>[entry.somedatefield.timestamp.add(86400000)]</pre> <p>to add 1 day to the date (value of 86400000 is in milliseconds)</p> <pre>[entry.somedatefield.timestamp.add(-86400000)]</pre> <p>to add -1 day (subtract a day) to the date (value of 86400000 is in milliseconds)</p> <p><i>i</i> Supports dynamic parameters via [entry.field_name]</p>	
<p>subtract(number)</p> <p>CORE</p> <p>CLOUD</p>	<p>Subtracts value from field value</p> <p><i>i</i> Supports dynamic parameters via [entry.field_name]</p>	
<p>multiply(number)</p> <p>CORE</p> <p>CLOUD</p>	<p>Multiplies value by given parameter value</p> <pre>[entry.somedatefield.multiply(2)]</pre> <p><i>i</i> Supports dynamic parameters via [entry.field_name]</p>	
<p>divide(number)</p> <p>CORE</p> <p>CLOUD</p>	<p>Divides value by given parameter value</p> <pre>[entry.somedatefield.divide(2)]</pre> <p><i>i</i> Supports dynamic parameters via [entry.field_name]</p>	
<p>split(separator)</p> <p><i>i</i> separator could be a space, like:</p> <pre>split()</pre> <p>CLOUD</p>	<p>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</p> <p>For example:</p> <p>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 into the desired representation</p> <pre>mytextfield.split().asArrayMultiSelect</pre> <p>Any other "array type" function can be applied. For example to get the count, one will write the following</p> <pre>mytextfield.split().asCount</pre>	
	<p>Now you can easily understand if the field value has been changed or not. Can be used ONLY (from ConfForms 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.</p> <p>Example in IFTTT macro:</p>	

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 **witho
ut** 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](#)

 Preview

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

Select macro

Save

Cancel

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
```

<p>asUserLink</p> <p>asUserLinks (same as above, but works on user multiselect fields)</p> <p>FROM V. 1.36.3</p> <p>Also, there is a function which renders it as a macro: useful for usage withing a template or ListView /CalendarView</p> <p>asUserLinkMacro</p>	<p>Generates a macro</p> <pre><ac:link><ri:user ri:userkey="USER_KEY_HERE"></ac:link></pre> <p>and renders the HTML out of it</p> <p>USER_KEY_HERE is looked up by function based on the username given</p>	<p>entry.myfield.asUserLink</p>
<p>asLink</p> <p>FROM V. 1.37.1</p> <p>CLOUD</p> <p>asLinks</p> <p>FROM V. 1.38</p> <p>CLOUD</p> <p>asLink(Some label)</p> <p>asLink({entry.somefield})</p> <p>FROM V. 1.44</p> <p>CLOUD</p>	<p>Returns value as HTML link (VALUE)</p> <p>Also, for collections: function "asLinks"</p>	<p>entry.myfield.asLink</p> <p>entry.myfield.asLink(I am a link label)</p>
<p>acLink</p> <p>CORE</p> <p>acLinkMacro</p> <p>CORE</p> <p>CLOUD</p>	<p>Renders a link to local resource</p> <p>acLinkMacro functions generates a macro only (without rendering to HTML)</p> <pre><ac:link><ri:page ri:content-title="SPACEKEY:PAGE_TITLE" /></ac:link></pre>	
<p>asImageLink</p> <p>CORE</p> <p>CLOUD</p>	<p>Renders link to given url or attachment Id (in cloud)</p>	<p>myfield.asImageLink</p>
<p>parseDate(FORMAT)</p> <p>FROM V. 1.36.6</p> <p>CLOUD</p>	<p>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</p> <p>Example on how to parse and format a date from JIRA issue</p> <pre>jirakey.fields.created.parseDate(yyyy-MM-dd'T'HH:mm:ss.SSSZ).formatDate(yyyy-MM-dd)</pre> <p>In this example we have a field called "jirakey" in ConfiForms Form, and access a property "created" from JIRA issue.</p> <p>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</p>	<p>entry.myfield.parseDate (JAVA_FORMAT)</p>
<p>toPersianDate</p> <p>CORE</p>	<p>Converts timestamp to Iranian/Persian date</p> <pre>date.toPersianDate</pre>	<p>date.toPersianDate</p>

<p>toPersianDateTime</p> <p>CORE</p>	<p>Converts timestamp to Iranian/Persian date with time</p> <pre>mydatefield.toPersianDateTime</pre>	<p>date.toPersianDateT ime</p>	<p>[</p>
<p>join</p> <p>FROM V. 1.38</p> <p>CLOUD</p>	<p>You can join the field values into one string. Works best on multi-select fields</p> <p>For example (expect a field to be a multi-select field here):</p> <pre>id:[entry.mymultifield.transform(id).join(OR id:)]</pre> <p>Will extract the ID's from a record stored, and then join (concatenate) those ID's into one string, separated by OR id:</p> <p>Something like this (when mymultifield contains 2 items):</p> <pre>id:1234-5678-9000-1234-5678-9000 OR id:1234-5678-9000-1234-5678-9001</pre>	<p>entry.myfield.. join(SOME VALUE)</p>	<p>[)]</p>
<p>asJIRAissue</p> <p>FROM V. 1.40</p> <p>CLOUD</p>	<p>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</p>	<p>entry.myfield. asJIRAissue. key</p> <p>entry.myfield. asJIRAissue. fields. customfield_x</p>	<p>/ is v J</p>
<p>asInsightObject</p> <p>FROM V. 1.49.1</p>	<p>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 accessed directy</p>		
<p>asPage</p> <p>FROM V. 1.40</p> <p>CLOUD</p>	<p>Helps you to transform the values saved to Page objects when used on Page/BlogPosts multi-select field</p> <p>Or to try to transform numeric value to page object (load page by id)</p>	<p>entry.myfield. asPage. spaceKey</p>	<p>/ c f</p>

<p>userInSecurityGroup securityGroupHas</p> <p>FROM V. 1.40</p> <p>CLOUD</p>	<p>Both functions allow you to check if a particular user belongs to a given security group or not</p> <p>Examples:</p> <p>If field "u" is holding a username and we want to check if this user belongs to "confluence-administrators" group</p> <pre>u.userInSecurityGroup(confluence-administrators)</pre> <p>If field "sg" holds a name of a security group and we want to check if a username is in this security group</p> <pre>sg.securityGroupHas(sash)</pre> <p>In the example below we check the same, but for currently logged in user</p> <pre>sg.securityGroupHas()</pre> <p>Both functions return "true" when condition is matched and "false" otherwise (without quotes)</p> <p>This means that if you need to put a filter to check the condition, the full expression will look like:</p> <pre>u.userInSecurityGroup(confluence-administrators):true</pre> <p>or</p> <pre>sg.securityGroupHas():true</pre>	
<p>securityGroups</p> <p>FROM V. 2.0</p> <p>CLOUD</p>	<p>Retrieves user's securityGroups in a comma separated list</p> <pre>userfield.securityGroups</pre> <p>(Assuming userfield is the field holding the username)</p>	
<p>get(index)</p> <p>FROM V. 1.42.4</p> <p>CLOUD</p> <p>Using get() without an index will return the last item from the list</p> <p>FROM V. 1.44.2</p> <p>CLOUD</p>	<p>Get's the element by index from a multi-value field</p> <p>To get the first element (index starts from 0)</p> <pre>somefield.get(0)</pre> <p>Another example with chaining the functions (when somefield holds a string value and we split it by "," and get the first element)</p> <pre>somefield.split(,).get(0)</pre>	

<p>append</p> <p>FROM V. 1.43</p> <p>CLOUD</p>	<p>Appends text to a value</p> <p>Will append a space to the value of "somefield"</p> <pre>somefield.append()</pre> <p>For multi-select dropdown, adding space after transformation and shows in as html link</p> <pre>somefield.transform(label.append()).asLinks</pre> <p> Argument can be dynamic, and reference another field, like [entry.somefield], so you can concat values together</p>	
<p>prepend</p> <p>FROM V. 1.45.2</p> <p>CLOUD</p>	<p>Prepends text to a value</p> <p>Will append a space to the value of "somefield"</p> <pre>somefield.prepend(SOME TEXT HERE)</pre> <p> Argument can be dynamic, and reference another field, like [entry.somefield], so you can concat values together</p>	
<p>lowerCase</p> <p>FROM V. 1.44.2</p> <p>CLOUD</p>	<p>Returns a lowercases value for the field</p>	
<p>upperCase</p> <p>FROM V. 1.44.2</p> <p>CLOUD</p>	<p>Returns a uppercased value for the field</p>	
<p>formatLinks</p> <p>FROM V. 1.45.2</p> <p>CLOUD</p>	<p>Tries to format the http(s) links found in the text as HTML links</p> <pre>somefield.formatLinks</pre>	

<p>greenhopperAsJSON</p> <p>FROM V. 1.45.3</p> <p>CLOUD</p>	<p>Tries to parse the field value returned by JIRA API for greenhopper fields</p> <pre>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]</pre> <p>to a structure you can access via properties</p> <pre>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</pre> <p>Example (to return a sprint name)</p> <pre>myJIRAFIELD.fields.customfield_10900.greenhopperAsJSON.name</pre>	
<p>asIFTTTRef (RESULT_NAME_AND_MORE)</p> <p>FROM V. 1.47.2</p> <p>CLOUD</p>	<p>The idea and the need for this function is the same as described for function <i>asEntryRef(REF_TO_ENTRY)</i></p> <p>This allows you to "escape" the <code>\$(iftttResult_NAME.someproperty)</code> into a function to workaround template evaluations against the current record. Useful when your ConfiForms Form creates a page with another ConfiFormForms Form and that form has various rules using <code>iftttResults</code> or <code>entry.fieldnames</code></p> <pre>entry.id.asIFTTTRef(MYRESULT.id) will produce \${iftttResult_MYRESULT.id}</pre> <pre>entry.id.asIFTTTRef(MYRESULT) will produce \${iftttResult_MYRESULT}</pre>	
<p>asUserProfileLink</p> <p>FROM V. 1.48</p>	<p>Outputs a field value as a link to user profile (with avatar and full name). The field must have a username as a value</p> <pre>[entry.myfieldholdingusername.asUserProfileLink]</pre>	

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

<p>asHex</p> <p>FROM V. 1.52.1</p> <p>CLOUD</p>	<p>Converts string into hex representation</p> <table border="1" data-bbox="329 178 532 352"> <thead> <tr> <th>string</th> <th>asHex</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>31</td> </tr> <tr> <td>2</td> <td>32</td> </tr> <tr> <td>hello</td> <td>68656c6c6f</td> </tr> </tbody> </table> <p>You can use it with other functions, as usual... something like</p> <pre>[entry.myfield.asHex.upperCase]</pre>	string	asHex	1	31	2	32	hello	68656c6c6f			
string	asHex											
1	31											
2	32											
hello	68656c6c6f											
<p>dec2Hex</p> <p>FROM V. 1.52.1</p> <p>CLOUD</p>	<p>Converts decimals to hex</p> <table border="1" data-bbox="329 564 532 779"> <thead> <tr> <th>value</th> <th>dec2Hex</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>As always, you can chain the functions</p> <pre>[entry.myfield.dec2Hex.upperCase] [entry.myfield.dec2Hex.trim(4)] - returns last 4 digits, instead of 000000D2 for 210 will return 00D2</pre>	value	dec2Hex	1	00000001	2	00000002	255	000000FF	210	000000D2	
value	dec2Hex											
1	00000001											
2	00000002											
255	000000FF											
210	000000D2											
<p>randomInt(minVal)</p> <p>FROM V. 2.0.8</p>	<p>You can generate tips from ConfiForms data with something like</p> <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> <p>To show "random" advice from your form of "advices"</p> <p>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)</p>											
<p>toString</p> <p>CORE</p>	<p>Ensures the value is a string value</p>											
<p>timezoneAwareDate</p> <p>CORE</p> <p>Since 2.13.7 supports setting the formatting pattern</p> <p>timezoneAwareDate (PATTERN)</p> <p>CLOUD</p>	<p>Tries to format the date given in the user's timezone. Uses user's defined formatting pattern to format the date.</p> <p>Unless given as parameter (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html)</p> <p>Date is always in the user's timezone</p>											

<p>timezoneAwareDateT ime</p> <p>CORE</p> <p>Since 2.13.7 supports setting the formatting pattern</p> <p>timezoneAwareDateT ime(PATTERN)</p> <p>CLOUD</p>	<p>Tries to format the date/time given in the user's timezone. Uses user's defined formatting pattern to format the datetime.</p> <p>Unless given as parameter (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html)</p> <p>Date is always in the user's timezone</p>										
<p>formatNumberWithLo cale(COUNTRY)</p> <p>CORE</p>	<p>Formats number with a given locale</p> <pre>formatNumberWithLocale(de) will format the number in German locale.</pre> <p>See supported locales in Java 8, https://www.oracle.com/technetwork/java/javase/java8locales-2095355.html</p>										
<p>compactDateTmeInt erval</p> <p>FROM V. 2.0.23</p>	<p>Helper function to render DateTme Interval field values in a more compact way when the date for start and end is within the same day</p> <pre>[entry.mydatetimeintervalfield.compactDateTmeInterval]</pre> <p>Example (the output format is specific to your Confluence date/time formatting settings):</p> <table border="1" data-bbox="329 884 1321 1213"> <thead> <tr> <th>DateTme Interval value</th> <th>Output</th> <th></th> </tr> </thead> <tbody> <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> </tbody> </table>	DateTme Interval value	Output		Mar 19, 2019 2:00 PM - Mar 19, 2019 3:00 PM	Mar 19, 2019 2:00 PM - 3: 00 PM	Transforms to a short format	Mar 19, 2019 2:00 PM - Mar 20, 2019 3:00 PM	Mar 19, 2019 2:00 PM - Mar 20, 2019 3:00 PM	Stays in long format, as event spans 2 days	
DateTme Interval value	Output										
Mar 19, 2019 2:00 PM - Mar 19, 2019 3:00 PM	Mar 19, 2019 2:00 PM - 3: 00 PM	Transforms to a short format									
Mar 19, 2019 2:00 PM - Mar 20, 2019 3:00 PM	Mar 19, 2019 2:00 PM - Mar 20, 2019 3:00 PM	Stays in long format, as event spans 2 days									
<p>truncWithExpand(N) where N is the number of characters to show</p> <p>FROM V. 2.0.25</p> <p>CLOUD</p>	<pre>[entry.field_name.truncWithExpand(10)]</pre> <p>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</p>										
<p>extractText</p> <p>CORE</p> <p>CLOUD</p>	<p>Extracts text from HTML value</p>										
<p>renderAsText</p> <p>CORE</p> <p>CLOUD</p>	<p>Render as Text (mainly to be used with values produced by wiki markdown field type)</p>										
<p>renderAsHtml</p> <p>CORE</p> <p>CLOUD</p>	<p>Render as HTML (mainly to be used with values produced by wiki markdown field type)</p>										

<p>renderWikiMarkup</p> <p>FROM V. 2.1.0</p>	<p>Renders wiki markup contents as html</p>	
<p>removeCRLFs</p> <p>FROM V. 2.1.0</p> <p>CLOUD</p>	<p>Removes all new lines in a field value</p>	
<p>removeSpaces</p> <p>FROM V. 2.1.0</p> <p>CLOUD</p>	<p>Removes all the spaces in the field value</p>	
<p>remove(value)</p> <p>CORE</p> <p>CLOUD</p>	<p>Removes value from the field (works as replaceWith for non-collections (single fields) and as remove item for multi-value fields)</p>	
<p>obscure</p> <p>FROM V. 2.6.0</p> <p>CLOUD</p>	<p>Hides the value behind the given mask</p> <pre>pwd.obscure(***** click to view)</pre> <p>Will create something like</p> <p>*****(click to view)</p> <p>Clicking on the "click to view" link will reveal the actual value of the "pwd" field</p>	
<p>hasValue(value)</p> <p>FROM V. 2.9.4</p> <p>CLOUD</p>	<p>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</p> <pre>field.hasValue(somevalue)</pre> <p>For single choice fields it checks for equality</p> <p>Can be used with Supported math operators, formulas and functions to construct conditional formulas</p> <p>For example:</p> <pre>IF([entry.Options.hasValue(choice1)], 10, 0)</pre>	
<p>escapeSQL</p> <p>FROM V. 2.9.5</p> <p>CLOUD</p>	<p>Escapes SQL parameters</p>	
<p>getOptions</p> <p>FROM V. 2.9.5</p>	<p>Allows you to get all the options registered with a field (choice based field, like radio group, checkbox group, dropdowns...)</p> <p>For example, to get the labels for options registered in a field called "radiogroup"</p> <pre>id.getOptions(radiogroup).transform(label)</pre>	
<p>getUnselectedOptions</p> <p>FROM V. 2.9.5</p>	<p>Allows you to get all the UNSELECTED options registered with a field (choice based field, like radio group, checkbox group, dropdowns...)</p> <p>To get unselected options from a field called "radiogroup" and show them on each line separately</p> <pre>id.getUnselectedOptions(radiogroup).transform(label.append(
))</pre>	

<p>queryAndRender (<formName: pageId>;<FILTER>; <FIELDS_OR_EXPRESSIONS>; <VIEW_TYPE>)</p> <p>FROM V. 2.10</p> <p>CLOUD</p>	<p>You can add a ConfiForms Field to your form's view and set it to show the contents of another form within your form</p> <ul style="list-style-type: none"> • <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: <ul style="list-style-type: none"> • table • card • list <p>Example</p> <pre>id.queryAndRender(f:819201;*;t;table)</pre> <p>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.</p> <pre>id.queryAndRender(myform:819202;partner:[entry.partner];name amount;card)</pre> <p>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".</p>	
<p>queryAndSet (<formName: pageId>;<FILTER>; <FIELD_OR_EXPRESSION>)</p> <p>FROM V. 2.16.9</p> <p>CLOUD</p>	<p>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</p> <p>Useful when you want to make a value lookup in ConfiForms IFTTT</p> <ul style="list-style-type: none"> • <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. <p>When multiple records much the filter the value will be concatenated</p> <p>Example:</p> <pre>id.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield)</pre> <p>same can be done though the _func helper</p> <pre>_func.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield) or [entry._func.queryAndSet(myform:1111222;myfield:[entry.field];anotherfield)]</pre>	
<p>pageProperties (property_name)</p> <p>FROM V. 2.10.7</p> <p>CLOUD</p>	<p>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)</p> <pre>mypage.pageProperties(property_name)</pre> <p>What is page properties: https://confluence.atlassian.com/doc/page-properties-macro-184550024.html</p>	
<p>sort(property_name ASC)</p> <p>sort(property_name1 ASC, property_name2 DESC)</p> <p>FROM V. 2.11.3</p>	<p>Sorts lists of choices / objects</p>	
<p>asUserTimezone</p> <p>FROM V. 2.12.4</p> <p>CLOUD</p>	<p>Represents given date as in user timezone date. Actually all 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.</p>	<p>someDate. asUserTimezon e.jiraDateTime</p>

<p>asUserTimezoneAwareDate</p> <p>asUserTimezoneAwareDateTime</p> <p>FROM V. 2.12.4</p> <p>Since 2.13.7 supports setting the formatting pattern</p> <p>asUserTimezoneAwareDate(PATTERN)</p> <p>asUserTimezoneAwareDateTime(PATTERN)</p> <p>CLOUD</p>	<p>Aliases for <i>timezoneAwareDate</i> and <i>timezoneAwareDateTime</i></p>	<p>Formats dates in user timezones</p> <p>someDate.asUserTimezoneAwareDate</p> <p>someDate.asUserTimezoneAwareDateTime</p>
<p>timezoneOffset</p> <p>FROM V. 2.13.2</p> <p>CLOUD</p>	<p>Returns offset between server and user's timezones in milliseconds</p>	
<p>htmlToWiki</p> <p>FROM V. 2.13.2</p> <p>CLOUD</p>	<p>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)</p>	
<p>iterateAndPrint</p> <p>FROM V. 2.13.10</p>	<p>Now it is possible to iterate through the multi-select fields and print their properties in a convenient way</p> <p>For example, iterating on multi user field and printing username and full name of the user in a row for each selected user</p> <pre>mymultiuserfield.iterateAndPrint([entry.username] - [entry.fullName]
).renderAsHtml</pre> <p>For comment field</p> <pre>mycomment.iterateAndPrint(<p>[entry.username] - [entry.timestamp.formatDate]: [entry.content]</p>).renderAsHtml</pre>	
<p>formatToPattern</p> <p>FROM V. 2.13.10</p> <p>CLOUD</p>	<p>Formats value to given regular expression pattern. For example, to format the phone number given 1234567890 into (123) 456-7890</p> <p>You will need to do something like this:</p> <pre>myfield.formatToPattern((\d{3})(\d{3})(\d+), (\$1) \$2-\$3)</pre> <p>Where first parameter is a grouping regular expression: (\d{3})(\d{3})(\d+)</p> <p>And the second one is how the "groups" should be printed (pattern itself): (\$1) \$2-\$3</p>	
<p>storageToHtml</p> <p>FROM V. 2.14</p> <p>CLOUD</p>	<p>Attempts to convert/render given value into HTML. Will work when the value is a valid storage format (Atlassian page storage format)</p>	

<p>evaluateFormula (FORMULA)</p> <p>FROM V. 2.17</p> <p>CLOUD</p>	<p>Evaluates and calculates a given formula. Formula should be given as explained here: Supported math operators, formulas and functions</p> <p>Parameters can be given as [entry.field_name] references</p> <p>Can be applied on a field or on _func for convenience</p> <pre>[entry._func.evaluateFormula(IF(EMPTY("[entry.somefield]", "ERROR", "ALL GOOD")))]</pre> <p>Or in ConfiForms Field macros it is easier to use via "id" field</p> <pre>id.evaluateFormula(IF(EMPTY("[entry.somefield]", "ERROR", "ALL GOOD")))</pre>	
<p>formatDuration (PATTERN)</p> <p>FROM V. 2.19.4</p> <p>CLOUD</p>	<p>This function helps to format numeric value in milliseconds as "duration"</p> <p>Example:</p> <pre>somefield.formatDuration(H:mm:ss) somefield.formatDuration(d H:mm:ss) somefield.formatDuration(d 'days' H:mm:ss)</pre> <p>Value of "somefield" needs to be numeric and represent time in milliseconds</p>	



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!**