Advanced Ratings & Calculation Syntax

Use this page to configure a "rating" sytem to evaluate tracker items or wiki pages. Introduced in Tiki5, the advanced rating feature allows for more control over the aggregation of scores. You will also see in this documentation page how to use the **calculations syntax**, which also applies at the Mathematical Calculation Tracker Field.

Related Topics

- E-democracy system
- Bugs and Wishes
- Rating

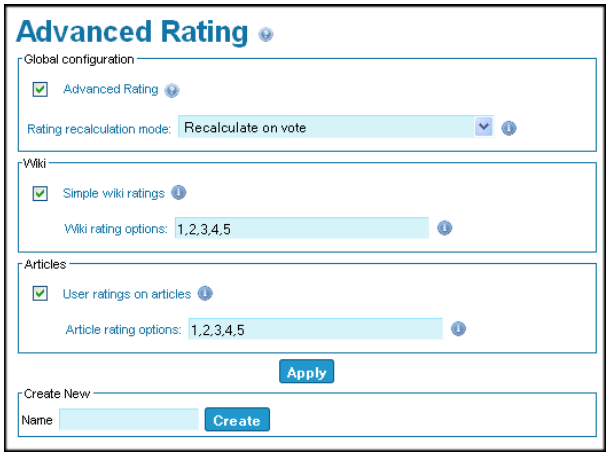
To access

Click the **Ratings** icon on the Admin Panel
or
Access **http://example.org/tiki-admin.php?page=rating**

Note

Tiki currently supports sorting through advanced rating in:

- Articles
- Wiki
- Comments
- See also Mathematical Calculation Tracker Field



Advanced Ratings page

Setting	Description	Default
Global configuration		
Advanced Rating	Enable the internal rating system, used for calculating values from trackers, articles, or other features.	

Setting	Description	Default
Rating recalculation mode:	Determines when and how rating aggregates are recalculated: * **On vote** (default): indicates that the score for the object should be recalculated every time a vote is performed. This option is suitable for sites with lower volumes and relatively simple calculation methods when ratings are used. * **Random on load**: will cause a few scores to be calculates on page load on a random basis (odds and count can be configured to adapt to site load). This option is suitable for calculation rules involving time that must be recalculated even if no new votes occurred. * **Random on vote** is similar to random on load, but will recalculate multiple scores (not necessarily including the current object) when a vote is performed. It is suitable for similar situations. The best option will depend on site load. * **Periodic**: is the best option for heavy load sites, making sure all calculations are done outside the web requests. A cron job must be set-up manually by the site's administrator. A sample script is available at the end of this page. Depending on the site load, some options may be better than others; on large volume sites, we recommend **cron job**. The **Recalculate on vote** recalculation may be inaccurate if rating calculation depends time. **Before any attempt to re-index the object:** Ties into the Search and List from Unified Index and updates the calculation at index-time.	Recalculate on vote
Recalculation odds (1 in X):		
Recalculation count:		
Wiki		
Simple wiki ratings	Enable a simple rating bar at the top of each wiki page.	
Wiki rating options:	List of options for the simple wiki ratings.	1,2,3,4,5
Articles	Enable a simple rating bar at the top of each articles page.	
User ratings on articles		
Article rating options:		

The feature must first be enabled through this same administration panel. Along with the feature, a few options are available. Among them, the score recalculation period must be defined. These are the available options:

- **On vote (default)** indicates that the score for the object should be recalculated every time a vote is performed. This option is suitable for sites with lower volumes and relatively simple calculation methods when ratings are used.
- **Random on load** will cause a few scores to be calculates on page load on a random basis (odds and count can be configured to adapt to site load). This option is suitable for calculation rules involving time that must be recalculated even if no new votes occurred.
- **Random on vote** is similar to random on load, but will recalculate multiple scores (not necessarily including the current object) when a vote is performed. It is suitable for similar situations. The best option will depend on site load.
- **Periodic** is the best option for heavy load sites, making sure all calculations are done outside the web requests. A cron job must be set-up manually by the site's administrator. A sample script is available at the end of this page.

For the random options, the odds of recalculating must be specified as a dice roll. For each occurrence of a recalculation, a limit to how many scores can be calculated must be specified to avoid the hang-up effect on the page load.

The value ranges for each object type can also be specified through the administration panels.

The common *sort_mode* parameter to lists can be used to activate sorting using advanced ratings. To do so, the sort mode must be set to *adv_rating_X_asc* or *adv_rating_X_desc* where *X* is the ID of the rating configuration. The default sort can also be set to advanced ratings in the administration panel where applicable.

Calculation configuration

From the administration panel, new calculations can be added. Initially, only the name is required. When created, the calculation will contain suitable default values.

For wiki pages:


Thus, visitors can provide feedback like:

- Did this page help you solve the issue?
- Was this page easy to understand?

Sorting items according to advanced rating

Note that the sort mode to use when needing to sort by advanced rating is either adv_rating_xx_asc or adv_rating_xx_desc, where xx is the ratingConfigId.

Set-up

By default, each calculated value is kept for 1 hour (3600 seconds). This limit does not apply when recalculating on vote, but is used for every other technique to avoid recalculating the same scores over and over again.

The calculation is defined as a small piece of code, similar to functional languages, which is very close to mathematical representations. Creating custom formulas is expected to require some mathematical skills. However, this documentation should provide examples for most frequent cases.

The editor in the administration panel performs extensive validation and will make it impossible to save the formula unless it can be evaluated. Checks are performed for:

- Syntax errors
- Unknown functions
- Missing arguments
- Invalid argument values
- Unknown input variables

Default formula

```
(rating-average (object type object-id))
```

It can be altered to limit the vote consideration to a limited time span, 30 days for example.

Recent votes only

```
(rating-average (object type object-id) (range (mul 3600 24 30)) )
```

In the language, spaces do not matter. Only the parenthesis indicate structure. **rating-average** is a function that fetches the ratings for a given object. *type* and *object-id* are standard variables fed when calculating a rating. **object** and **range** are configuration options of the function.

mul is a mathematical function. (mul 3600 24 30) is equivalent to 3600*24*30.

The functions can be combined in various ways. For example, we could calculate a score that considers the votes from the past month, but gives extra emphasis on the recent ones.

Combined vote duration

```
(add (rating-average (object type object-id) (range (mul 3600 24 30))) (rating-average (object type object-id) (range (mul 3600 24 7))) )
```

Even though the votes are 1-5, the final score can be on an entirely different scale. The language is also extensible if the calculation needs to be combined with other factors or weight. See Rating Language.

All available options are documented in the following section.

Syntax

General Reference

Sample and use case

add (Sum)

Performs a simple sum accepting multiple input.
It is possible to chain the calculation with several values.

Simple addition

```
(add 3 4) -> 7 (add (add 3 4) 5) -> 12 (add 3 4 5) -> 12 (add 4 0.5) -> 4.5
```

The calculation can be done directly using tracker field permaname for the same tracker.

Using tracker field permaname

```
permaname_1 value = 1 permaname_2 value = -1 permaname_3 value = 4 (add permaname_1 permaname_2) -> 0 (add (permaname_1 permaname_2 permaname_3) -> 4
```

sub (Substract)

Performs a simple substraction accepting multiple input

Examples

```
(sub 3 4) -> -1 (sub (sub 3 4) 5) -> -6 (sub 3 4 5) -> -6 (sub 4 0.5) -> 3.5
```

div (Divide)

Performs a simple division accepting multiple input values.

Examples

(div 3 4) -> 0.75 (div (mul 3 10) 5) -> 6 (div 30 5 3) -> 2 (div 4 0.5) -> 8

mul (Multiply)

Performs a simple multiplication accepting multiple input values.

Examples

(mul 3 4) -> 12 (mul (mul 3 4) 5) -> 60 (mul 3 4 5) -> 60 (mul 4 0.5) -> 2

and / or

and

Ensures all elements evaluate to true.

Examples

(and 3 2 1 2 3) -> 1 (and 2 3 0 2) -> 0

or

Ensures that at least one element evaluates to true. Elements are evauated sequentially until a false element is found. Others are left unevaluated.

Examples

(or 3 2 1 2 3) -> 1 (or 2 3 0 2) -> 1 (or 0 0) -> 0

avg

Calculates the average of multiple values. All entries in the list will be flattened if arrays are present.

Examples

(avg 1 2 3) -> 2 ... given list contains [1, 2, 3] (avg list) -> 2

Note: Unless told otherwise `avg` treats empty values and zeros the same, but you can use `(if)` to check like this:

Empty example

(if (not (equals calcValue_3 "")) (avg calcValue_1 calcValue_2 calcValue_3) (if (not (equals calcValue_2 "")) (avg calcValue_1 calcValue_2) (if (not (equals calcValue_1 "")) calcValue_1)))

avg can be used together with other fields values as well as an item-list field displaying values from another tracker.

Calculate the average of values from a different tracker in an item-list

The field permanameListofnumbers contains values 1 and 3 (grabbed from tracker permanameNumbersfromanother tracker) (avg (for-each (list permanameListofnumbers) (formula (concat permanameNumbersfromanother tracker)))) -> 2

clean

Clean accents and special characters from a string of text, replacing the accented characters with the non-accented equivalent where possible. Added in Tiki 18.2

Examples

(clean (str foó barça caña)) -> "foo barca cana"

coalesce

Returns the first non-empty value from the list.

Examples

(coalesce 3 4) -> -3 (coalesce (sub 3 3) 5) -> 5 (coalesce 0 0 (str) -10) -> -10 (coalesce 0 0 0 0 0) -> 0

comment

Any comment block is stripped from the formula at parse-time

Examples

(mul 1 2 (comment Simple enough?)) -> 2

concat

Concatenates a string of text. (new in Tiki12)

Note: The quoted string syntax was included in Tiki13.

Examples

(concat (str $) 1234) -> "$1234" (concat 14 (str %)) -> "14%" (concat 14 "%") -> "14%"

contains

Searches for a value within a vector of values (new in Tiki15.0, backported to Tiki14.2 and Tiki12.5).

Examples

(contains 4 (1,2)) -> 0 (contains 4 (2,4)) -> 1

Up to Tiki 25: Note that if you are using an argument value in here, you would need to eval and then put brackets around to make it work.

args.values_by_permname.version: 305,306

(contains 307 (eval(args.values_by_permname.version))) -> 0 (contains 305 (eval(args.values_by_permname.version))) -> 1 (contains 30 (eval(args.values_by_permname.version))) -> 0

Tiki 26+: arguments don't need to be eval-ed anymore. The previous example has this syntax now:

args.values_by_permname.version: 305,306

(contains 307 args.values_by_permname.version) -> 0 (contains 305 args.values_by_permname.version) -> 1 (contains 30 args.values_by_permname.version) -> 0

count

Returns the total number of entries within an array passed as argument.

Examples

(count results) -> 5

You can use it with split-list to count the number of items in a list or in an item-link (should work with dynamic items list).

Counting item in item-link

Let say I have an item-link permanameBooks with 12 items linked. (count (split-list (content permanameBooks) (separator ,) (key a))) -> 12

Only one key need to be created it will populate properly

Currency

New in Tiki21: https://sourceforge.net/p/tikiwiki/code/71175

Allows to convert a calculation into currency field. Expects 3 arguments. First is the calculation of the amount. Second is the source currency - i.e. which currency is the amount in? Third is the currency field.

Examples

(currency (cal-of-the-amount) (sourceCurrency) currencyFieldPermName)

If you use the Canadian daily exchange table from the Bank of Canada you can retreive sourceCurrency string from currencyFieldPermName (ie: FXUSDCAD) using this formula:

Get the last 3 char of a value in a tracker

(substring currencyFieldPermName -3)

Examples

(currency (cal-of-the-amount) (substring currencyFieldPermName -3) currencyFieldPermName)

currency-convert

New in Tiki22: https://sourceforge.net/p/tikiwiki/code/76059

This adds 2 things:

1. currency-convert math function which will allows us to convert specific math functions to CAD when you need only CAD values.
2. using default tracker for exchange rates when parsing a math function output like 123USD without a currency field context. The tracker must contain 'exchange rate' in its name in order to be found. This will be until we have a proper concept of system trackers.

Date and Time

Note about Date and Time and timezone

Time stored in the DB or in the Search Index is always UTC. (converted based on the Tiki settings)

Tiki convert back the value to display based on the Tiki preferences or the user preferences settings related to timezone.

Calculation are done on the value (unix timestamp from the database) in UTC and date formulas have to include this factor to allow exact calculation.

For example, for a server with a timezone set to Paris (UTC+1) the following formula will return today's date with the time of the field "momentDate" -1 hour (Difference between Europe/France timezone and UTC).

Wrong

(str-to-time (concat (date (str Y-m-d)) (date (str H:i) momentDate)))

To allow the calculation to be performed on the stored time in the database we need to specify that it is UTC. The following formula will return today's date with the time of the field "momentDate" as displayed.(and expected)

Right

(str-to-time (concat (date (str Y-m-d)) (date (str H:i) momentDate) (str UTC)))

Tiki will know to convert back the value to display it following Tiki preferences.

Date

New in Tiki14.1: https://sourceforge.net/p/tikiwiki/code/55964

Takes two optional arguments, format and timestamp and uses the PHP **date** function. See here for reference. Format defaults to "U" which is the Unix timestamp, and the timestamp defaults to "now".

Date Examples

(date) > Returns "1438358437" currently (date (str Y-m-d H:i:s)) > Returns "2015-07-31 17:00:43" currently (date (str r) theTimeAndTheDate) > Returns "Tue, 16 Jun 2015 09:23:09 +0100" for instance, where theTimeAndTheDate is the permanent name of a tracker field in the same tracker.

Date-diff

New in Tiki26

Calculates the difference between two dates as unix timestamps. Optionally supply a third parameter to make it calculate the difference between two dates calculating only working days. See str-to-time for more info about calculation of working days.

equals

Compares multiple values.

Examples

(equals 2 (add 1 1) (sub 4 2)) -> 1 (equivalent of 2 == 1+1 && 2 == 4-2) (equals (add 1 1) 3) -> 0

for-each

For a list of value pairs, such as the output of *split-list*, evaluates a formula for each set of values, returns the list of results.

Within the formula, variables coming from the list will be used first. Fallback will be on the other variables available in the execution context.

As of Tiki18, list items can be the output of ItemsList tracker field where individual formula fields are the linked fields from the other tracker. See example below.

Examples

... given items contains [{a: 1, b: 2, c: 3}, {a: 2, b: 3, c: 4}] (for-each (list items) (formula (mul a b c))) -> [6, 24] ... given items contains [{a: 1, b: 2, c: 3}, {a: 2, b: 3, c: 4}] ... and d contains 10 (for-each (list items) (formula (mul c d))) -> [30, 40] ... given trackerDetails is an ItemsList field pointing to another tracker with a field detailsAmount ... and particular tracker item has 2 linked items with detailsAmount = 30 and 40 (add (for-each (list trackerDetails) (formula (concat detailsAmount)))) -> 70 This formula sums the detailsAmount column from the other tracker for all items linked from this tracker's item.

hash

Generates a hash based on multiple values. Used primarily to generate aggregate hashes in the PluginActivityStream. Note that because it is a hash, the exact value coming out does not matter. Only that given the same parameter, it will produce the same value.

Examples

(hash 1) -> [sha1("1")] (hash 1 2 3 4) -> [sha1("1/2/3/4")] (hash 1 2 (map (a 3) (b 4))) -> [sha1("1/2/3/4")]

if

Conditionally evaluates a branch.

Examples

(if (equals 2 2) 42 -1) -> 42 (if (equals 2 1) 42 -1) -> -1

You can use it with STR to apply a condition on a string within another field.
IE: if permanameFieldName contain "Bernard"

If used with a string to show another field

(if (equals permanameFieldName (str Bernard)) permanameFieldMoney (str 0)) -> Will output the value of permanameFieldMoney if true (permanameFieldName = Bernard) -> Will output 0 if this is not true (permanameFieldName ≠ Bernard)

IsEmpty

New in Tiki14: https://sourceforge.net/p/tikiwiki/code/53588

Examples

(IsEmpty 1) -> 0 (IsEmpty 0) -> 1

You can also use a tracker field permaname as true or false value or as returned value from the tracker item.
Here we combine if, equals and is empty to create a different output for each case else the default output (0 or 1) will be returned.

(if (equals 1 (IsEmpty permaname)) (str empty) (str notempty)) if the field has no value (1) -> empty if the field has a value (0) -> notempty

IsEmpty may also be used to test if an array is empty.
However in some case you may have an error. coalesce seems to work better for testing and displaying tracker field values.

less / more

less-than

New in Tiki14.1 and Tiki12.5: Checks whether the first value is less than the second.

Examples

(less-than 3 (add 2 2)) -> 1 (less-than (add 2 2) 3) -> 0

more-than

New in Tiki14.1 and Tiki12.5: Checks whether the first value is more than the second.

Examples

(more-than 3 (add 2 2)) -> 0 (more-than (add 2 2) 3) -> 1

lower

Converts string to lower case.

Examples

(lower Foo) -> foo

map

Generates a map (or dictionary).

Examples

(map (key1 1) (key2 2) (key3 (str value3))) -> {"key1": 1, "key2": 2, "key3": "value3"}

Not

New in Tiki14: https://sourceforge.net/p/tikiwiki/code/53590

number-format

Format a number with grouped thousands. See PHP's number_format function for exact arguments. New in Tiki18.

Examples

(number-format 123456.78) -> "123,456.78" (number-format 120.334 (str 3)) -> "120.334" (number-format 120.334 (str 0)) -> "120" (number-format 123456.78 (str 2) (str ,) (str)) -> "123 456,78"

pad

New in Tiki15 committed in r57094

Takes two to four arguments, input string, output length, padding string (defaults to a space) and padding type (defaults to right) See http://php.net/manual/en/function.str-pad.php for more info.

This example right pads prices in a simple products tracker for sorting and filtering

(pad productsPrice 8 (str 0) (str left))

round

Rounds to a specific number of digits (new in Tiki12)

Examples

(round 4.556234342234 2) -> 4.56 (round 4.556234342234) -> 5

str

Generates a static string when needed and the processor attempts to process the string as a variable. Any arguments will be concatenated using spaces.

Note: The quoted string syntax was included in Tiki13.

Examples

(str hello-world) -> "hello-world" (str hello world) -> "hello world" (str hello world foobar) -> "hello world foobar" (str (mul 2 3) "= 6") -> "6 = 6" (str some text with new line~nl~) -> "some text with new line\n"

The following is useful to add a space between calculation or values from different sources (tracker field value)

To add a space

(str) (str hello)(str world) -> "helloworld" (str hello)(str)(str world) -> "hello world"

str-to-time

Parse about any English textual datetime description into a Unix timestamp. See PHP's strtotime function for more details on valid formats and options. New in Tiki18.

Examples

(str-to-time (str 2017-06-05)) -> "1496610000" (date (str Y-m-d) (str-to-time (str +1 day) (str 2017-05-29))) -> "2017-05-30"

Numbers of days between a date in a field and now

(round (div (sub (str-to-time permanameDate) (date)) 86400) 0)

Working days example

(str-to-time (str +10 working days) (str-to-time (str 2024-12-20)))

New in Tiki 26: in order to make use of business/working days calculation, you should create a Holidays calendar, add the event dates that are holidays (might be one-time, single-day, multi-day events or recurring events like every weekend Sat/Sun, for example). Then, you can choose this holidays calendar in Tiki admin calendar section to be the default holidays calendar. Then, any calculation involving 'business' or 'working' days will actually exclude any days marked as holidays.

str-replace

Replace substring in a string. See PHP's str_replace function for exact arguments. New in Tiki18.

Examples

(str-replace (str foo) (str bar) (str hello foo)) -> "hello bar"

preg-replace

Perform a regular expression search and replace on a field value. The preg-replace math function is available in Tiki 21+

Using preg-replace to return part of a string before a | (pipe

The field permanameMonthAndUser contain "June | Bernard" (preg-replace (str /\|.*$/) (str) permanameMonthAndUser) -> "June"

Using preg-replace to return part of a string after a | (pipe

The field permanameMonthAndUser contain "June | Bernard" (preg-replace (str /^.*?\|/) (str) permanameMonthAndUser) -> "Bernard"

split-list

Produces a multi-dimensional array out of a text string. Each line is expected to be an independent value, each line will be split by a separator into the specified keys.

Examples

... given str contains a list of 3 comma-separated values (split-list (content str) (separator ,) (keys a b c)) -> [{a: 1, b: 2, c: 3}, {a: 2, b: 3, c: 4}]

subtotal

Special function to aggregate data in a table. See Report formatting for more details.

upper

Converts string to upper case.

Examples

(upper Bar) -> BAR

Advanced Rating-specific Reference

rating-average and rating-sum

The rating functions calculate the score from the rating history table. Each rating performed on the site is kept in the database and can be used to calculate custom ratings on. The various options allow to adapt the score calculation to reflect the importance on the site, whether it is to support quality improvement on documentation or to rank incoming data on a feed aggregator.

- **object**, mandatory and always *(object type object-id)* in this context.
- **range**, to limit how long votes are considered. Argument is provided as a number of seconds.
- **ignore**, with *anonymous* as an argument to only consider votes from registered users.
- **keep**, to only consider one vote per visitor. Unless the option is present, all of the votes are taken into account. The option can be either *latest* or *oldest* to indicate which one to keep.
- **revote** can be specified if **keep** is specified. Indicates the time period required between votes. For example, users could be allowed to vote more than once per day, but only their latest vote each day would be considered, if revote is set to mul(24 3600). If the user voted yesterday as well as today, both votes will be counted.

article-info

Pulls information from an article to include in the calculation. The first argument must always resolve to 'article'. If any other value, the calculation will be skipped for the evaluated object, making the formula type-specific.

Available properties:

- rating, the static rating attached to the article
- view-count
- age-second
- age-hour
- age-day
- age-week
- age-month

Examples

(article-info type object-id rating) (article-info (str article) 42 age-month)

attribute

Pulls information from the generic object attributes.

Examples

(attribute (object type object-id) (property tiki.proposal.accept)) -> [value for page in a rating calculation] (attribute (object (str wiki page) 14) (property tiki.proposal.accept) (default 0)) -> [value for page id 14]

tracker-field

Pulls information from the tracker item. The field value is converted to numeric value automatically. Zero is provided if the value is not found or unapplicable.

Examples

(tracker-field (object (str trackeritem) 42) (field priority)) -> [value contained in the tracker item field with permanent name ''priority'' from tracker item with object Id 42]

You can pull the value of an item coming from an item link and an item dynamic list field type (not tested on item list). This imply using the permaname of the item link tracker field and the permaname of the field that contain the value (in the other tracker).

value form item link item

(tracker-field (object (str trackeritem) permaname_thistrackerfield) (field permaname_othertrackerfield))

category-present

Gives 1 in score of every listed category present on the object.

Examples

(category-present (object type object-id) (list 3 4)) -> [0, 1 or 2 - Depending on how many of categories 3 or 4 are on the object]

Appendix

When unified search is used, recalculation can be configured to be done during re-indexing, removing the need for this script.

Cron job

<?php chdir('/path/to/tikiroot'); require_once 'tiki-setup.php'; require_once 'lib/rating/ratinglib.php'; $ratinglib->refresh_all();

Sample and use case

Calculate date difference to see if tracker item is new or not

In this use case we compare a field that contains creation date to the actual date and if the difference is lower than 7 days (604800) the field will have the value "new" else "notnew".

Examples

```
(if(equals (more-than 604800 (sub (add (date) 0) (add dateDeCrAtion 0))) 1) (str new) (str notnew))
```

Using Concat to create a reference made of month and string value from other fields

In this use case we concatenate the month as word from a date field and a value from a text field (can be a dropdown, etc) and we use to create a reference for this item. Then the result, the value, can be used for many things like populating an itemList field, filtering, searching, etc.

Filling a field with value for type and month

```
(concat trackerPermanameType (str " | ") (date (str F) trackerPermanameDate))
```

For example it will set the value of field with : "Credit | February"

Add different values from fields from a Tracker in another Tracker

In this use case we have 2 trackers. One that records time spent (permaname_timeSpent) with individuals for different interaction with an individual (tracker actions) that contain an itemLink to the other tracker. And the other tracker (tracker users) that will collect all the actions and the time spent for this user and calculate (display) the total time spent per type of actions.

We first use an itemList (permaname_listActionsAndTime) to filter all the actions made for this specific user as you will usually do : ItemList => TrackerB, TrackerB_usernameField, TrackerA_usernameField, field to display : Actions and Time, use the format that fit the fields order to display and easy to read list of actions|time.

You should have a list displaying in each tracker user items something like that:

```
Meeting | 15 Meeting | 20 Phone | 5 Administrative | 60 Documentation | 60 etc...
```

Anywhere after it in the list fields (orders matters with calculation fields) for this tracker (tracker users) create a mathematical field.
Now you need to check all the rows of the itemList and if it contain a specific actions add the time value from the other tracker.

We want to see the the time spent **with** the user so we will use a condition (if) based on the string corresponding to the action and a OR statement to collect the time spent if the permanameActions equals a specific string.
If there is no time the field output will be "0".

```
(add (for-each (list permaname_listActionsAndTime) (formula (if (or (equals permanameActions (str Meeting)) (equals permanameActions (str Call)) ) permaname_timeSpent 0) ) ) )
```

Assign a label to a range of values

In this case we combine less-than and more-than in a cascade of conditions to determine what value (a label) should be saved in the field based on a numerical value from another existing field. For example we have appartments and a field with their surface and we want to group them by size (less than 80m2, between 80m2 and 120m2, etc...)

If used in a search (customSearch, pluginList filter, etc) the label must be unique; Tiki search engine (tested with MySQL search) will consider characters like "_", "-" , "+" (may be more) as separators and not character of the string. For example if you search for "20" both "10-20" and "20-30" will be outputted in the results.

(if (less-than trackerPermanameSize (add 79 0)) (str 79) (if (less-than trackerPermanameSize (add 119 0)) (str 80-119) (if (less-than trackerPermanameSize (add 159 0)) (str 120-159) (if (less-than trackerPermanameSize (add 199 0)) (str 160-199) (if (less-than trackerPermanameSize (add 239 0)) (str 200-239) (if (less-than trackerPermanameSize (add 279 0)) (str 240-279) (if (more-than trackerPermanameSize (add 279 0)) (str 280))))))))

To add some safeties we can also test first if there is a value in the field.
Here the idea is to check if the trackerPermanamePrice contain a value (price) then to group the items within the same range of prices by giving them a label.

(if (equals 1 (IsEmpty trackerPermanamePrice)) (str) (if (less-than trackerPermanamePrice (add 99999 0)) (str 99999) (if (less-than trackerPermanamePrice (add 299999 0)) (str 100000-299999) (if (less-than trackerPermanamePrice (add 499999 0)) (str 300000-499999) (if (less-than trackerPermanamePrice (add 699999 0)) (str 500000-69999) (if (more-than trackerPermanamePrice (add 699999 0)) (str 700000)))))))

Simple Wiki Ratings

Related

- Rating
- Rating Revamp
- Mathematical Calculation Tracker Field
- Grouped Data

alias

Advanced+Rating | AdvancedRating | Advanced Ratings | AdvancedRatings