

# Attribute Value Formats

Attribute values are formatted automatically by the pimatic frontend. Generally, the idea is to present attribute values in a better, human-readable way. The formatting depends on various aspects such as the attribute value type, the unit type associated with a values (if defined), and the size of the value. In some cases the resulting display of the value is not what the user would like to see. For example, for currency values the user may want to have the value displayed with a fixed number of digits (for cents) rather than having the display of decimals depending on the actual value. For this reason, it is possible to override the default format with a format selected by the user.

## xAttributeOption

name

*Name for the corresponding attribute.*

displaySparkline  No *not set*  
*Show a sparkline behind the numeric attribute*

hidden  No *not set*  
*Hide the attribute in the gui*

displayFormat

*Override formatting conventions used by the GUI to display the value. Format outputs are: raw, fixed, localeString, and uptime*

User-defined display formats are set by defining a xAttributeOption for the respective attribute as part of the device configuration. The easiest way of doing this is to use the device editor of the pimatic frontend. As shown in the screenshot a xAttributeOption is linked to an attribute exposed by the device. If you need to find out which attributes are exposed use the Variables editor of the pimatic front to get a list of all variables including device attributes. The "displayFormat" xAttributeOption use a simple syntax as follows:

- Id of the formatter to be applied, one of `uptime | fixed | localeString`

## Attribute Value Formats

- An optional list of comma-separated options: `[, <name>: <value>]*` . String value need to be provide in quotes. Number and Boolean value are given without quotes
- Example: `fixed, digits: 2`

## uptime

---

The uptime formatter is meant to be used for numeric attributes providing time duration values given in seconds. The formatter will produce a string presenting human-readable breakdown of the duration into applicable time unit, similar to what the Linux command `"uptime -p"` produces. The following options are provided:

- `hideUnit` : Whether or not the unit type shall be hidden. Set to `true` by default.
- `format` : One of `"long"`, `"medium"`, `"short"` . Set to `"long"` by default.
  - `"long"` : Displays day(s) / hour(s) / minute(s) / seconds(s) where values equal to 0 are omitted
  - `"medium"` : Omits display of second(s), unless the uptime value is smaller than 1 hour
  - `"short"` : Only displays day(s) and hour(s) , if value is smaller than 1 day the values is displayed as hh:mm:ss
- `units` : One of `"long"`, `"short"` . Set to `"long"` by default.
  - `"long"` : Display long units, i.e. day(s), hour(s), minute(s), seconds(s). If locale is set the local specific mapping is displayed
  - `"short"` : Display abbreviated units, i.e, d, h, min, s

## fixed

---

The fixed formatter is meant to be used for numeric attributes to formats number using fixed-point notation. By default the number will be formatted with no decimals in place. The following options are provided:

- `digits` : The number of digits to appear after the decimal point (default: 0), for example: `digits: 2` .

## localeString

---

The localeString formatter is meant to be used for numeric attributes to provide the value using locale conventions. Basically, it uses the capabilities of the Javascript [*toLocaleString* function for *Number* objects]([https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Number/toLocaleString](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number/toLocaleString)). The following, optionas are applicable:

## Attribute Value Formats

- `locale(s)` : A string with a BCP 47 language tag, for example: "de\_DE"
- `localeMatcher` : The locale matching algorithm to use. Possible values are "lookup" and "best fit"; the default is "best fit"
- `style` : The formatting style to use. Possible values are
  - "decimal" for plain number formatting,
  - "currency" for currency formatting, and
  - "percent" for percent formatting; the default is "decimal"
- `currencyDisplay` : How to display the currency in currency formatting. Possible values are
  - "symbol" to use a localized currency symbol such as €, "code" to use the ISO currency code,
  - "name" to use a localized currency name such as "dollar"; the default is "symbol"
- `useGrouping` : Whether to use grouping separators, such as thousands separators or thousand/lakh/crore separators. Possible values are true and false; the default is true
- `minimumIntegerDigits` : The minimum number of integer digits to use. Possible values are from 1 to 21; the default is 1
- `minimumFractionDigits` : The minimum number of fraction digits to use. Possible values are from 0 to 20; the default for plain number and percent formatting is 0; the default for currency formatting is the number of minor unit digits provided by the ISO 4217 currency code list (2 if the list doesn't provide that information)
- `maximumFractionDigits` : The maximum number of fraction digits to use. Possible values are from 0 to 20; the default for plain number formatting is the larger of `minimumFractionDigits` and 3; the default for currency formatting is the larger of `minimumFractionDigits` and the number of minor unit digits provided by the ISO 4217 currency code list (2 if the list doesn't provide that information); the default for percent formatting is the larger of `minimumFractionDigits` and 0
- `minimumSignificantDigits` : The minimum number of significant digits to use. Possible values are from 1 to 21; the default is 1
- `maximumSignificantDigits` : The maximum number of significant digits to use. Possible values are from 1 to 21; the default is 21