

---

**trytond\_product**

*Release latest*

**unknown**

**May 29, 2023**



# CONTENTS

<b>1 Usage</b>	<b>3</b>
1.1 Using product templates and variants	3
1.2 Categorising products	3
<b>2 Configuration</b>	<b>5</b>
2.1 price_decimal	5
2.2 uom_conversion_decimal	5
<b>3 Design</b>	<b>7</b>
3.1 Product	7
3.2 Category	8
3.3 Unit of Measure	8
3.4 Unit of Measure Category	8
3.5 Configuration	9
<b>4 Release notes</b>	<b>11</b>
4.1 Version 6.8.0 - 2023-05-01	11
4.2 Version 6.6.0 - 2022-10-31	11
4.3 Version 6.4.0 - 2022-05-02	11
4.4 Version 6.2.0 - 2021-11-01	11
4.5 Version 6.0.0 - 2021-05-03	12
4.6 Version 5.8.0 - 2020-11-02	12
4.7 Version 5.6.0 - 2020-05-04	12
4.8 Version 5.4.0 - 2019-11-04	12
4.9 Version 5.2.0 - 2019-05-06	13
4.10 Version 5.0.0 - 2018-10-01	13
4.11 Version 4.8.0 - 2018-04-23	13
4.12 Version 4.6.0 - 2017-10-30	13
4.13 Version 4.4.0 - 2017-05-01	13
4.14 Version 4.2.0 - 2016-11-28	13
4.15 Version 4.0.0 - 2016-05-02	14
4.16 Version 3.8.0 - 2015-11-02	14
4.17 Version 3.6.0 - 2015-04-20	14
4.18 Version 3.4.0 - 2014-10-20	14
4.19 Version 3.2.0 - 2014-04-21	14
4.20 Version 3.0.0 - 2013-10-21	14
4.21 Version 2.8.0 - 2013-04-22	15
4.22 Version 2.6.0 - 2012-10-22	15
4.23 Version 2.4.0 - 2012-04-24	15
4.24 Version 2.2.0 - 2011-10-24	15

4.25	Version 2.0.0 - 2011-04-26 . . . . .	15
4.26	Version 1.8.0 - 2010-11-01 . . . . .	15
4.27	Version 1.6.0 - 2010-05-09 . . . . .	15
4.28	Version 1.4.0 - 2009-10-19 . . . . .	16
4.29	Version 1.2.0 - 2009-04-20 . . . . .	16
4.30	Version 1.0.0 - 2008-11-17 . . . . .	16

The *Product Module* defines the essential concepts needed to describe products in Tryton. It lets you define the basic properties of products and their variants, and allows you to record how the quantity of a product is measured.



The items found under the *[Product]* main menu item allow you to view and manage the products on your system.

## 1.1 Using product templates and variants

Tryton makes it easy to create *Products* which have one or more slightly different *Variants*.

A simple example of where you could use this is in a company that supplies clothes. Each type of shirt may come in a range of different sizes, but most of the properties of the shirt, such as the name, unit of measurement, categories, and so on, would be the same. So in this case you would create a single *Product* to represent the shirt, and a *Variant* for each of the different sizes. Structuring your products like this can help you manage and update them.

---

**Tip:** You may find that your products are not suited to being structured in this way. If so, don't worry, when you create a new *Product* from the *[Products → Products]* menu item a single variant is automatically created for you and displayed as part of the product.

---

## 1.2 Categorising products

It can be a good idea to organise your *Products* into groups. This makes it much easier to find and manage them effectively, especially if you have a lot of products on your system.

The product *Categories* are designed for this purpose. You can create categories with any name you want, and then add the appropriate categories to each product. Each product can belong to as many, or as few, categories as required. The categories can also be organised into a structure, with each category having a parent category and some subcategories. This can help you classify your products more finely.

For example, a clothes supplier may use these categories:

```
Accessories
Clothes
  Shirts
    Short sleeves
    Long sleeves
  Jumpers
Range
  Spring Summer
  Autumn Winter
  Christmas
```

Based on these categories, you may decide that a particular lightweight shirt belongs in the Clothes / Shirts / Short sleeves and Range / Spring Summer categories.

---

**Tip:** To get a list of all the products in a category first open the [*Products* → *Categories*] menu item. Then when you open one of the categories listed here you will get a list of all the products in that category.

---



## CONFIGURATION

The *Product Module* uses some settings from the [product] section of the [configuration file](#).

### 2.1 price\_decimal

The `price_decimal` setting defines how many decimal places are used when storing *Products'* unit prices.

**Warning:** Once the database has been created you cannot reduce this value, doing so will break your system's data integrity. Also if you want to increase this value you must also manually change it in the database *IR Configuration*.

The default value is: 4

### 2.2 uom\_conversion\_decimal

The value from the `uom_conversion_decimal` setting defines the number of decimal places used when storing the conversion rates and factors between *Units of Measure*.

**Warning:** Once the database has been created you cannot reduce this value, doing so will break your system's data integrity. Also if you want to increase this value you must also manually change it in the database *IR Configuration*.

The default value is: 12



The *Product Module* introduces the following concepts:

## 3.1 Product

The *Product* is the main concept introduced by the *Product Module*. It represents an item or service that Tryton stores information about, and is often something that gets bought, sold, or produced.

In Tryton product definitions are made up from two parts, the *Product Template* and the *Product Variant*. Both of these are sometimes referred to as the *Product*, depending on the context.

Each product template can have many different product variants, but each product variant is based exclusively on one product template.

A product's full code is made up from a prefix code which is defined on the product template and a suffix code which is specified on the product variant. The full product code can then be used as the product's **SKU**.

### 3.1.1 Product Template

The *Product Template* defines the set of properties common to a group of *Product Variants*. These properties include things like its list price, its type, its code, what *Categories* it is in, and what *Units of Measure* are used by default for quantities of the product.

**See also:**

A list of product templates can be found by opening the main menu item:

*Products* → *Products*

### 3.1.2 Product Variant

Each *Product Variant* inherits many of its properties from its *Product Template*. It does, however, have some properties that are specific to each variant such as the description, cost price, and suffix code which is important to distinguish between different variants.

The identifiers that are used to refer to a product are also specific to a product variant. A product identifier is made up from a type and a code.

Some of the supported types of identifier include:

- International Article Number (EAN)
- International Standard Audiovisual Number (ISAN)

- International Standard Book Number (ISBN)
- International Standard Identifier for Libraries (ISIL)
- International Securities Identification Number (ISIN)
- International Standard Music Number (ISMN)

**See also:**

A list of all the product variants is available from the main menu item:

*Products → Products → Variants*

## 3.2 Category

The product *Category* concept provides a flexible way of grouping *Product Templates* together. The categories can be structured by giving them a parent category and some sub-categories.

**See also:**

A list of product categories can be found by opening the main menu item:

*Products → Categories*

## 3.3 Unit of Measure

The *Unit of Measure* concept provides the units by which the quantity of a *Product* is measured. These are things like, meter, mile, kilogram, hour, gallon, and so on.

Each unit of measure belongs to a *Unit of Measure Category*.

Quantities can be converted to a different unit of measure from the same category using the unit of measures' rates or factors. It is also possible to specify the rounding precision and number of decimal digits used when rounding or displaying values from the unit of measure.

**See also:**

The units of measure can be found using the main menu item:

*Products → Units of Measure*

## 3.4 Unit of Measure Category

A *Unit of Measure Category* is used to group together *Units of Measure* that are used to measure the same type of property. These are things like length, weight, time or volume.

**See also:**

The units of measure can be found using the main menu item:

*Products → Units of Measure → Categories*

## 3.5 Configuration

The product *Configuration* contains the settings which are used to configure the behaviour and default values for things associated with products.

There are configuration options for the sequences to use to automatically generate codes for *Products*.

**See also:**

The product configuration can be found using the main menu item:

*Products → Configuration → Configuration*



## RELEASE NOTES

### 4.1 Version 6.8.0 - 2023-05-01

- Bug fixes (see mercurial logs for details)
- Remove support for Python 3.7
- Add support for Python 3.11
- Remove non-standard work day unit
- Add energy units and category
- Add barcode function on product identifier
- Add identifier\_get to product

### 4.2 Version 6.6.0 - 2022-10-31

- Bug fixes (see mercurial logs for details)

### 4.3 Version 6.4.0 - 2022-05-02

- Bug fixes (see mercurial logs for details)
- Forbid to decrease digits of unit of measure
- Add support for Python 3.10
- Remove support for Python 3.6

### 4.4 Version 6.2.0 - 2021-11-01

- Bug fixes (see mercurial logs for details)
- Prevent editing list and cost prices without a company
- Make list price not required on product
- Provide digits from unit of measure
- Allow only fixed cost price as method for service

## 4.5 Version 6.0.0 - 2021-05-03

- Bug fixes (see mercurial logs for details)
- Deactivate variants when product is deactivated
- Allow multiple templates to be added/removed on a category

## 4.6 Version 5.8.0 - 2020-11-02

- Bug fixes (see mercurial logs for details)
- Remove support for Python 3.5
- Allow the product template code to be set from a sequence
- Set code column first in template tree view
- Add default order on product and template
- Check price\_decimal configuration

## 4.7 Version 5.6.0 - 2020-05-04

- Bug fixes (see mercurial logs for details)
- Add round\_price
- Make product code unique
- Add code to template as variant prefix code
- Hide product fields inherited from template when empty
- Skip default products when template is created from product
- Search also on variant name when searching on template record name

## 4.8 Version 5.4.0 - 2019-11-04

- Bug fixes (see mercurial logs for details)
- Allow the product code to be set by a sequence
- Allow UoM conversion between different categories
- Add product identifiers



## 4.9 Version 5.2.0 - 2019-05-06

- Bug fixes (see mercurial logs for details)
- Remove starting wildcard when searching on codes and numbers

## 4.10 Version 5.0.0 - 2018-10-01

- Bug fixes (see mercurial logs for details)
- Remove support for Python 2.7

## 4.11 Version 4.8.0 - 2018-04-23

- Bug fixes (see mercurial logs for details)
- Enforce digits length on Rounding Precision on Unit of Measure

## 4.12 Version 4.6.0 - 2017-10-30

- Bug fixes (see mercurial logs for details)
- Show all products by category
- Limit cost method to fixed for service
- Move cost\_price from Template to Product

## 4.13 Version 4.4.0 - 2017-05-01

- Bug fixes (see mercurial logs for details)
- Add ceil and floor as uom rounding

## 4.14 Version 4.2.0 - 2016-11-28

- Bug fixes (see mercurial logs for details)
- Add relate from Template to Products

## 4.15 Version 4.0.0 - 2016-05-02

- Bug fixes (see mercurial logs for details)
- Replace category by categories on Template
- Add Python3 support
- Add configuration for default cost price method
- Change Uom.round into an instance method
- Add Function fields on Product from Template

## 4.16 Version 3.8.0 - 2015-11-02

- Bug fixes (see mercurial logs for details)
- Be more strict on unit of measure conversion

## 4.17 Version 3.6.0 - 2015-04-20

- Bug fixes (see mercurial logs for details)
- Add support for PyPy
- Ensure that uom rounding is always greather than zero
- Add price\_digits
- Add parameter price\_decimal

## 4.18 Version 3.4.0 - 2014-10-20

- Bug fixes (see mercurial logs for details)

## 4.19 Version 3.2.0 - 2014-04-21

- Bug fixes (see mercurial logs for details)

## 4.20 Version 3.0.0 - 2013-10-21

- Bug fixes (see mercurial logs for details)

## 4.21 Version 2.8.0 - 2013-04-22

- Bug fixes (see mercurial logs for details)

## 4.22 Version 2.6.0 - 2012-10-22

- Bug fixes (see mercurial logs for details)

## 4.23 Version 2.4.0 - 2012-04-24

- Bug fixes (see mercurial logs for details)
- Remove special uom search clause
- New product types: Goods, Assets and Service
- Category is no more required on product

## 4.24 Version 2.2.0 - 2011-10-24

- Bug fixes (see mercurial logs for details)

## 4.25 Version 2.0.0 - 2011-04-26

- Bug fixes (see mercurial logs for details)

## 4.26 Version 1.8.0 - 2010-11-01

- Bug fixes (see mercurial logs for details)

## 4.27 Version 1.6.0 - 2010-05-09

- Bug fixes (see mercurial logs for details)
- Improve computation precision of UOM conversion

## 4.28 Version 1.4.0 - 2009-10-19

- Bug fixes (see mercurial logs for details)

## 4.29 Version 1.2.0 - 2009-04-20

- Bug fixes (see mercurial logs for details)
- Allow egg installation

## 4.30 Version 1.0.0 - 2008-11-17

- Initial release