
tryton*currency*

Release latest

Aug 21, 2021

Contents

1	Setup	3
1.1	Loading and updating currencies	3
2	Usage	5
2.1	Setting currency exchange rates	5
2.2	Scheduling rate updates	5
3	Configuration	7
3.1	rate_decimal	7
4	Design	9
4.1	Currency	9
4.2	Rate	9
4.3	Scheduled Rate Updates	9
5	API Reference	11
5.1	Monetary field	11
	Python Module Index	13
	Index	15

The *Currency Module* defines the basics needed to work with currencies in Tryton including a list of standard currencies. It is also possible to define the rates used to convert amounts between different currencies.

On activation, the module does not create any currency records. It is possible to load them from the ISO database.

1.1 Loading and updating currencies

There is a script called **trytond_import_currencies** that creates and updates *Currencies*.

You can run it with:

```
trytond_import_currencies -c trytond.conf -d <database>
```


In order to perform conversions between currencies exchange rates must be defined.

2.1 Setting currency exchange rates

All the *Currency* values are relative. Best practice is to define which currency you want to use as the base currency by setting its exchange *Rate* to 1 for the date when it is first used. Then the rates of all other currencies are set as a multiplier of the base currency.

In order to set an exchange rate of $1.0 \text{ EUR} = 1.1222 \text{ USD}$ for the 1st of January the following records should be created:

- A rate of 1.0 for the EUR (Euro) currency with the 1st of January as the date.
- A rate of 1.1222 for the USD (US Dollar) currency with the 1st of January as the date.

If you then wanted to update the exchange rate to $1.0 \text{ EUR} = 1.1034 \text{ USD}$ for the 15th of January you just need to set the rate on the USD currency to 1.1034 for that date.

Note: In this example, as there isn't any rate set for the dates between the 2nd and 14th of January the last available rate will be used. Here this rate would be 1.1222 as this was the rate set for the 1st of January.

2.2 Scheduling rate updates

You may want to define some *Scheduled Rate Updates* to keep the *Exchange Rates* up to date. When doing this you have to pick a source, the base *currency* and a frequency. Then for each required date since the last update, a rate of 1.0 will be set for the base currency and the corresponding rate will be set for each of the selected currencies.

Warning: It is strongly advised to use the same base currency for all *Scheduled Rate Updates*.

Note: The currency module only supports the rates provided by the [European Central Bank](#), but third party modules can add additional sources for exchange rates.

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

3.1 rate_decimal

The `rate_decimal` setting defines how many decimal places are used when storing currencies' *Rates*.

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: 6

The *Currency Module* introduces the following concepts:

4.1 Currency

Each currency of interest is represented in Tryton by a currency record. These currencies are used together with a numeric value to represent an amount of money. Tryton can convert monetary amounts from one currency to another using the exchange *Rate* for a given date. This is important for multi-currency transactions, and if you want to manage money in a range of different currencies.

See also:

Currencies can be found by opening the main menu item:

Currency → *Currencies*

4.2 Rate

The exchange rates that are used when converting money between *Currencies* are saved next to their associated currencies. As exchange rates vary over time they are stored along with the date from when they apply. All exchange rates are relative with respect to each other.

4.3 Scheduled Rate Updates

The Exchange Rates can be automatically updated using a Scheduled Task. When the scheduled task is run it uses the external source to fill in the missing rates since the last update. The frequency determines which dates get rates.

See also:

Scheduled Rate Updates can be found by opening the main menu item:

Currency → Scheduled Rate Updates

5.1 Monetary field

class `trytond.modules.currency.fields.Monetary` (*string* [, *currency* [, ***options*]])

A subclass of `Numeric` used to store monetary values.

`Monetary.currency`

The name of the `Many2One` field which stores the currency used to display the symbol.

t

`trytond.modules.currency.fields`, [11](#)

C

currency (*trytond.modules.currency.fields.Monetary*
attribute), 11

M

Monetary (*class in trytond.modules.currency.fields*), 11

T

trytond.modules.currency.fields (*module*),
11