
Product Attribute Strict Documentation

Release stable

April 13, 2015

1	Key differences	3
2	FAQ	5
3	Authors and Contributors	7
4	Professional Support	9

Tryton has a powerful system of attributes for products introduced by the `product_attribute` module. The module however uses the `fields.Dict` (JSON in database) field type to store attributes and this reduces the consistency since the lack of foreign keys and constraints allow deletion, renaming and editing of selection values even when used by products.

This module tries to remain as close as possible to Tryton's attribute system, but avoids the dictionary field to have attributes stored on separate table.

Key differences

1. Options of attributes with the *Selection* type are stores in a new table. (Core module stores that in a text field in a JSON like syntax)
2. Attributes of product are stored on a separate table (Core module stores this as JSON serialized text in database)

FAQ

1. Can you move data from the tryton core module to this module ?

Yes, you can. An example script is provided to migrate the values from Tryton's default attribute system to this one. The script can be found under `scripts/migrate_from_core_module.py`

2. Can this module be installed alongside core 'product_attribute' module ?

Nope. You can use only one of the two modules and obviously we recommend ours

Authors and Contributors

This module was built at [Openlabs](#).

Professional Support

This module is professionally supported by [Openlabs](#). If you are looking for on-site teaching or consulting support, contact our [sales](#) and [support](#) teams.