Import Best Practices

Sources: compiled from myodoo.de, best practices import, data migration decisions
Odoo web resources: Import Data User Doc, Export Data User Doc

Introduction

Data migration is one of the tasks that could take a lot of time during a project's implementation and can therefore be a major cause of delay to reach a first go-live. It is important to anticipate that to guarantee that the project can be implemented on time.

There are two types of data to take into consideration when migrating data:

- Master data, like contacts & products
- Transaction data, like invoices, sale orders or purchase orders.

The master data must exist in the system to be able to create transaction data.

When discussing data migration with your customers, a lot of questions will arise and important decisions must be made. Following are some of the most important topics to tackle to make sure that this part of the implementation can be implemented in the smoothest & fastest way.

TOC

General Concepts

Master data vs transactional data

Master data

Master data contains all information that is assumed to play a key role in the core business of a company. Master data can contain data about customers and clients, employees, products, suppliers, analyses and more. Master data is typically shared by multiple users and groups within an organization.

Master data is static, which means that it rarely changes and is usually valid in the long term. Since master data is used by several areas of a company, it is highly relevant for all business processes. Furthermore, transaction data is dependent on master data. No transaction data exists without master data. For these reasons, master data is generally held on a long-term basis.

Typical master data are for example product data, suppliers, customers, employees, locations or inventory.

Transactional data

In the context of data management, transactional data is the information collected from transactions. Transactional data can be of a financial, logistical, or work-related nature and can range from an order, to shipping status, to hours worked, to insurance costs, to claims.c

Transactional data is therefore dynamic data that is used by individual departments and whose relevance is only limited to a certain time.

Typical transactional data could be orders, stocks, purchase orders or invoices. Because transactional data is dynamic, it quickly becomes historical data that is mainly used for reporting.

Why do we focus on master data?

Because master data is static (unchanging over long periods of time) and has broad uses across the company, focusing on importing this data is often more beneficial and efficient than focusing on importing transactional/historical data. Master data (opening balance, initial inventory, products, contacts, etc.) is essential for the business, but most transactional data is not.

Most historical data does not have real added value; specifically, transactional/historical data is usually needed only for traceability purposes. Therefore, the time it takes to import this data is not worthwhile.

Importing data vs. creating data manually

Importing through a file always seems to be faster than importing manually, but this statement might be wrong in many cases, and each one must be analyzed case by case. To decide which method to use to import the data, here are some factors to take into account:

The amount of records VS the amount of time to create the file

Sometimes the customer can't get a file that has the same format as the file we would need to import. It could be because the previous system has a different structure, or because the format of the exported data cannot be manipulated (pdf instead of excel sheets). If it takes as much time to create the file as it would to create the data manually, then it is best that the SPOC uses the system directly to get used to the way it works.

The affinity of your customer with Excel

This goes along with the previous point. If your customer is unable to provide a decent file because of a lack of skills with excel, then you should simply change the strategy and see how you could reduce the amount of manipulation in excel. For instance, if you have imported variants with a lot of attributes and combinations, the file can quickly become messy and impossible to import. By just importing the master data (product template, attributes, variant) and asking your customer to manually add the variants to the product, you might actually speed up the whole process rather than trying to do everything through an import.

The complexity of the data to import

Importing a contact or an invoice is totally different. The import template is much more complicated for the invoice and there might be more fields to add; it could mean having to define a process to import each piece of information sequentially. Also, some record information might be more complicated to import (many2many fields compared to text fields). Maybe those fields could be added manually or updated through an action.

Open documents at the go-live

When your customer starts using Odoo, they already have some open orders, invoices, purchase orders, deliveries... It is not necessary to import all this information at the moment of the go-live. You will get more details in the next section.

How should we migrate the transactional data at go live?

Preparing the excel file to import SO/PO/Invoices takes a lot of work and time. The risk is that your client might miss the deadline because he is not ready with all the files despite it not being necessary to have everything in the system at that moment. What is most important is that the customer knows how to handle every aspect that is covered by the go-live. If they can do that, then they will be able to manually create their documents (Sale orders, Purchase orders, Invoices...) when they need them.

For example: The go-live for the purchase application is scheduled in one month. You can suggest that when they create a purchase order in their existing system, they also create the document in Odoo without confirming it. At the moment of the go live, they will have all the purchase orders of the past month. Then you can do the following things depending on each documents:

- Remove it if it has been invoiced and delivered
- If it has been delivered but not invoiced yet, you will remove it and create the invoice directly in the accounting module
- If it has been invoiced but not yet delivered, you will remove it and create the receipts in the inventory module
- If no delivery and invoicing have been done, then you can simply process the purchase order normally

This is a good exercise for your customer, it reduces the pressure and stress at the go-live. There are other options to manage the import of the data, for instance to only create them once needed after the go live, following the same ideas as above.

But all data cannot be imported this way because you will probably need it once you start working. If you need to manage the inventory at the go-live, not having your stock in the system will be very complicated. Again, it is fine if part of the information is not accurate or in the system but you will need a solid basis of data to work efficiently. That's the kind of import that customers should spend time on.

Data cleansing and import responsibilities

When working with the client so they can cleanse their own data, the BSA's role is to:

- Ask for an exhaustive list of the fields that should be imported, analyze what information should be kept, and determine how to map the customer information into the Odoo structure
- Create an Excel file containing a template for Odoo import
- Create an instructional document or video containing the rules to follow in order to populate the Excel file. Ex: a short video explaining the vlookup to populate the ID's and map child contacts with their parent company
- Send these files to the client. They will populate the Odoo import template with their data according to the rules provided in the instructional document.

When importing the client's data, the BSA's role is to:

- Move the Excel file (either one you've created, cleansed, and tracked changes on, or one they provided in the correct format) to a shared folder so that the BSA and client are both working off the same document.
- Train the client to manage the import themselves instead of doing it for them. This is because we want the client to update data or create new data whenever they want.

Making the right choices about importing data

When is it okay to import data for the customer and when should the client be autonomous?

Even if the BSA has good Excel knowledge, it is not the BSA's responsibility to perform data cleansing. This is the responsibility of the customer.

The more experience you gain, the faster it will be for you to clean your client's spreadsheets. But be careful to not go too far and take too much responsibility in this task. You always have to find a balance between wanting to arrange the file yourself in place of your SPOC and delegating the fixing and cleaning to the SPOC.

First of all, you need to assess the technical knowledge of your SPOC: are they able to understand Excel & some technical aspects of Odoo? The worse their technical knowledge, the more you will be involved in this process. The more technical they are, the more likely you are to fix the error faster (use of IDs, matching of records through vlookup...). On the contrary, non-technical/basic errors such as a bad spelling or lower/upper case issues can be delegated to your SPOC.

The aim is to see which tasks can easily be assigned to your client and which tasks require further analysis. The potential time savings should also be taken into account. The consultant should clean the data if the issues are technical and if the time saved is significant. If, on the other hand, the data cleaning task will take considerable time no matter who does it, it may be better to leave it to the client and not waste this time on the package. Once again: it's all about balance.

The one thing you can never endorse the responsibility for is when it comes to business decisions. You cannot decide if a product will go in a specific category, if a contact is linked to a specific company, on which financial account an amount should be inputed... The reliability and accuracy of the data is always the SPOC responsibility.

If the client has limited resources or knowledge when it comes to data cleansing, and the client specifically asks the BSA to clean and import the data, communicate that this work will be billed.

Cleaning the data for the client

If the client still wants the BSA to clean the data, go ahead!

When importing data for a client, it is important to keep in mind that any data cleansing could affect the integrity of the data. One way to ensure correct data integrity is to keep a list of every change made to the dataset.

How to challenge the historical data import

It may happen that your client wants to import all the historical data from their previous system when beginning with Odoo. You should avoid that as much as possible because:

- it's generally not easy
- it can be very time consuming
    - *Ex: Your customer wants to import all the invoices for the last 5 years. In order to import this information, you'll have to import all the contacts first and some may not exist in their actual list of contacts. So you will have to create those old (and useless) contacts. This may be the same for the products. Then for each invoice you may have several lines that you will also have to import which is not easy to format in a spreadsheet. Furthermore, the invoices need to be validated, marked as paid and reconcile with the correct amount, that it will impact the accounting reports... you will probably work during dozens of hours on data that will rarely be accessed instead of working on improving the way the company operates moving forward.*
- it represents a huge workload both on your side and the customer's side.

Most of the time this wish to import all the historical data is driven by the fear of losing something that could be useful one day.

Why is it a risk for the project?

Importing old data into the database is always time-consuming as there are always very complex models that are involved to create those data.

The time wasted to import historical data may vary a lot and will depend on:

- The output given by the customer: what file has he provided, the amount of records to be imported, the match between the data exported and what Odoo will need to import the data

- The complexity of the model where the data must be imported: the more models and the more related fields there are, the more complex the import will be
- Your own skill level and the client's skill level with excel to be able to quickly correct the file
- If the client has time to export data

How can I convince my customer not to import historical data?

The idea is to convince the client that importing historical data will not be efficient and that it is better to focus on more relevant points such as focusing on workflows and their implementation to manage the future of their activity. Legacy data decreases in value quickly over time; they are not a source of revenue and if reporting can be done on those, it can probably be done outside of Odoo.

If we come back to our example here are some tips to convince the SPOC to not import all the historical data: "Maybe the previous system can still be running for a while, even if it is only in read access so they can still get access to the information. If the data can be exported and kept on an excel file, it might be sufficient to retrieve the important information and to do some reporting. Moreover importing the invoices will increase the import complexity of the opening balances as the amount already booked in Odoo will have to be removed from the amount of the imported journal items".

To be better prepared to challenge and convince your customer to not import those data, try asking the following questions:

- What is the added value of having this data in Odoo compared to using them in a separate system?
- If this data is imported, what use will your customer make of them? How often are they going to consult them? For what purpose?
- Is the data he wants to import still relevant today (data from several years ago is less and less relevant over the years)? Has the data changed over the years? Maybe only a part of the data could be imported (only the last year, month...)
- Is the reporting generated from the import of those data really relevant?

In short, always challenge the usefulness of the data your customer needs to import: if historical data is needed for reporting purposes s/he could keep it in an Excel file outside the system. If s/he needs to compare the historical data to the current data in Odoo, s/he could export the data from Odoo into an Excel file and compare in Excel.

Why is it an error to focus on cleaning master data before doing the import?

In an implementation, we don't want to delay the training or go live because of master data cleaning. If the customer's files are too dirty or would take too much time to clean, it is best practice just import what the client has (always import only relevant information) and move on with the implementation. If in the future the client has the time to clean up their files, they can do so using an existing Excel file from their previous imports of relevant records since we have generated ID's. Odoo can then replace the cleaned master data.

Should the customer clean all of their data?

Switching to a new ERP feels like an opportunity for the customer to clean up all the data coming from their old system. But in reality, the customer wouldn't need to do that if he was able to work with the data before (he would have cleaned it up already).

Cleaning data is not a good idea because not all of it is critical; maybe only 10% of the contacts are really active and the rest are almost never used. Focusing on cleaning everything is then a risk of delay with very

limited or no added value being gained from this task. Information will always change and learning to update or clean the information in Odoo is more relevant and will actually be a good exercise for the customer. And by working in Odoo, if the SPOC realizes that there is a lot of wrong information, it can also be easily fixed through an update by exporting the data from Odoo, correcting in Excel, and importing the changes back into Odoo.

V14 has introduced a data cleaning application that could actually save a lot of time since the data will be imported and then merged together. It will give you additional reasoning when explaining to your customer why you want to avoid wasting time cleaning the files before doing the import.

Cleaning data also requires knowing what is not clean. That implies adding an additional person to this task, trying to get everyone ready at the same time, which is sometimes quite complicated. The less people involved, the better to avoid wasting time importing data.

Data Import Tips & Tricks

Use the external ID

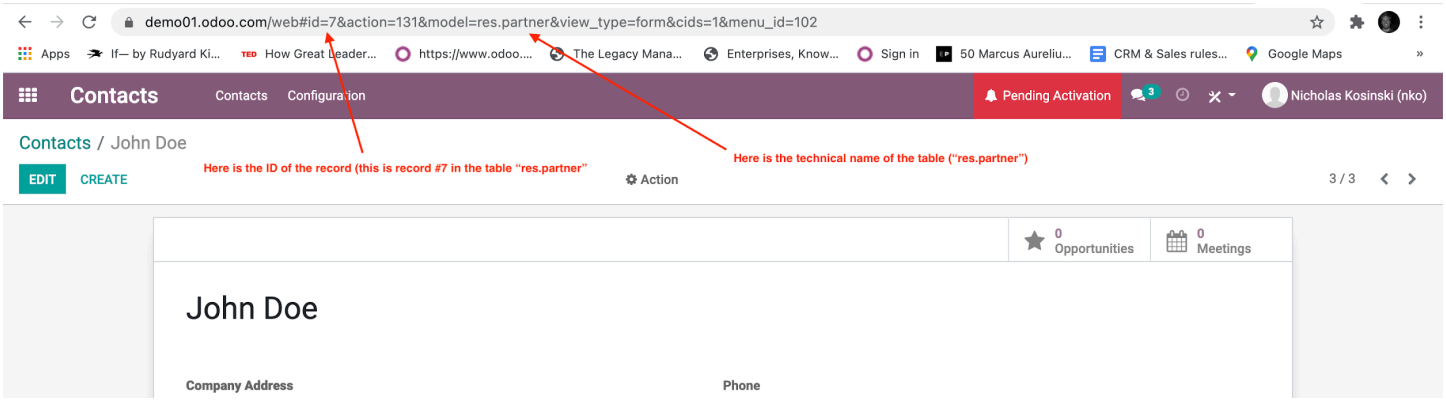
When a record is imported into Odoo with no external ID, no external ID is generated. It is not until the first export that an external ID is created. When importing data to create records, you should always create an external ID in order to easily update your data in the future. Any updates that need to be made to a record must include this external ID; if no external ID is listed on the update spreadsheet, Odoo will create new records instead. therefore it is best practice to create an external ID upon import to allow for easier updating in the future. See this video here for an explanation of how to update records using the external ID: See Video Here

External ID vs Internal ID

In Odoo, each record has an external ID which is a unique identifier throughout the whole database. When creating an external ID for initial import, we use the format below: model, BSA's trigram, a number.

`Product_template_atm_00001`

When records are imported, an internal id is also generated by Odoo which is unique in the table (the one in the URL) but not across the entire database.



When I export the record shown above (I am only exporting 2 fields: Name and External ID), this is what I will see in excel:

res.partner (1)

Display Name	External ID
John Doe	__export__.res_partner_7_3807debe

Note: when an external ID is automatically created (if no ID was created on import) the format is as pictured above.
When this record is opened in Odoo, in debug mode go to 'View Metadata' to see both the external ID (listed as XML ID) and the internal ID (listed as ID):

Metadata

ID:	7
XML ID:	__export__.res_partner_7_3807debe
No Update:	false (change)
Creation User:	apt@odoo.com
Creation Date:	02/18/2021 09:55:04
Latest Modification by:	apt@odoo.com
Latest Modification Date:	02/18/2021 09:55:04

When you import records into the system with a user generated External ID Odoo will automatically add a prefix of "__import__" separated from the External ID by a period. For example, if you upload a record with an External ID of "398430" upon export the entire External ID will be "__import__.398430". When a record has a prefix of "__import__" the prefix is not needed when importing updates to the record.

When you import records into the system without an External ID Odoo will only generate a unique External ID upon the export of those records. Any system generated External IDs will contain a "__export__" prefix. When updating a record with a "__export__" prefix you are required to keep the prefix so Odoo can match the record with the existing record. If you remove the "__export__" prefix a new record will be created instead of the intended update to an existing record. The new record would have the same base ID containing a prefix of "__import__".

The prefix simply dictates the direction in which the External ID was created, on import or export. The prefix field and the external identifier is stored in the system inside the ir.model.data model. You can find a list of all external identifiers by navigating to Settings / Technical / Sequences & Identifiers / External Identifiers (debug mode required). The combination of the prefix and external identifier must be unique across the entire database.

Fill the fields, fill the models this is hard

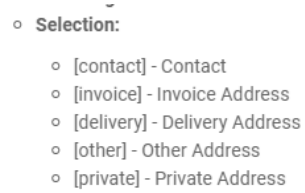
General

Odoo's data is case-sensitive: lowercase letters and uppercase letters are not the same character on the script of import. That means that "professor" is not equal to "Professor" and will not be recognized by Odoo if the actual value is "Professor". Don't forget, if you use an external id, you will avoid these issues.

The column title should correspond to the technical field, not its label name. Odoo can recognize the label name, but it is best practice to use the technical name to mitigate risk of importing data into the wrong field: sometimes two different fields can have similar or identical label names. In order to know the technical name of the field and/or the values of a selection field, activate the debug mode and hover over the field. You can also export the field in order to display its technical name on a spreadsheet.

Selection fields

Selection fields have values and labels. Data can be imported using either the values or the labels. Ex: the contact address field lists both when hovering over an address option: [value] - [label]



Many to One (m2o) fields

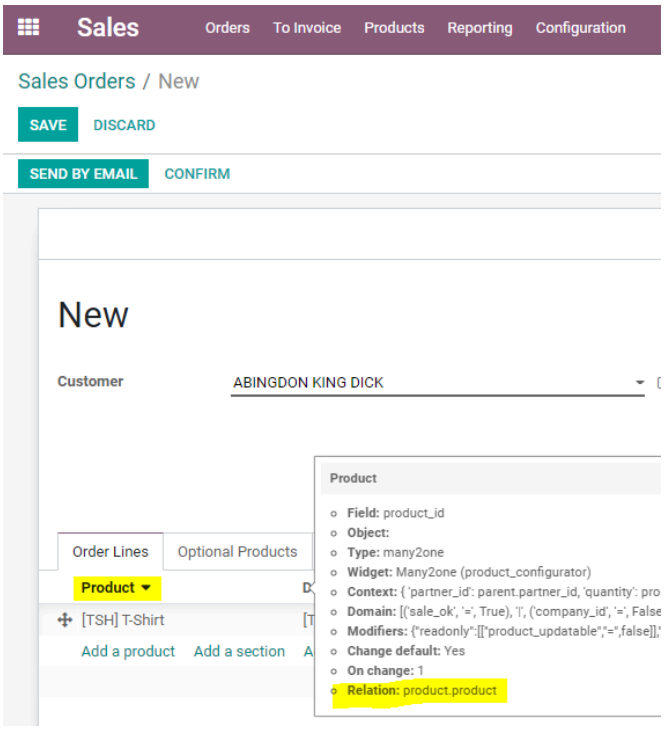
Create all records first before relating them to each other. When importing, use the external IDs for all records as these are the only fields guaranteed to be unique (unlike the record name which could be the same for multiple records).

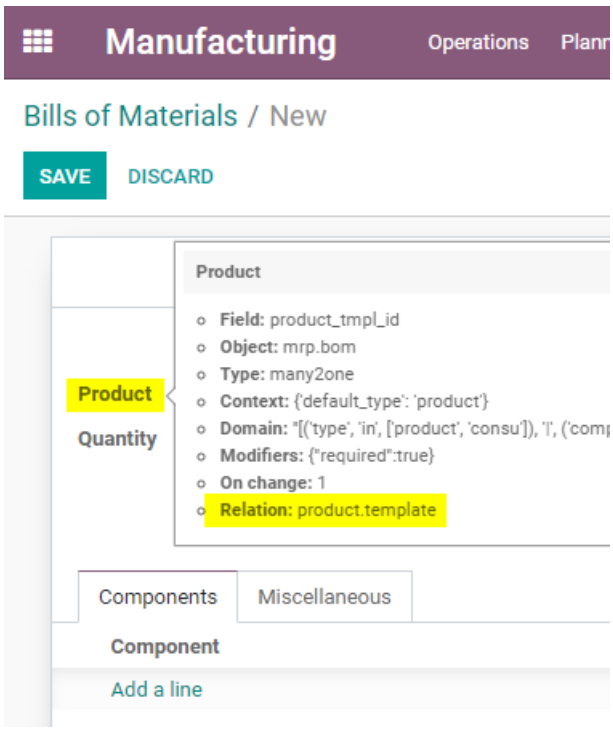
Imagine you have created two companies called ‘Odoo’, and you want to relate a contact to one of the companies both named ‘Odoo’. Using the name field when importing this relationship means you have a 50% chance of linking to the wrong company.

Another example: the locations (for the inventory adjustment). Indeed, if the customer has the locations WH1/Stock and WH2/Stock → you can’t write this name on the excel file, Odoo will not recognize it. The actual name is Stock for both locations. This means the external ID is the only way to differentiate these locations.

Products and their variants

If the external id is used to import data including the product (ex : bill of material, inventory adjustment, ...) then be careful about the model: product.template vs product.product** Indeed, the same product has 2 different records in 2 different models, with 2 different external ids. In order to know which model the field is related to, activate debug mode, hover over the field, and check the relation.





Importing products is tricky as there are 2 external IDs: the one corresponding to the model product.template and the one corresponding to the model product.product **.

You should import the product templates then either export the template list or configure the variants (model product.product)

Be careful when you import data including the product (ex : bill of material, inventory adjustment, …) : in order to know which model the product field is related to, activate the debug mode, set the mouse on the field and check the relation.

*** Why do we have 2 different models for products? Product.template corresponds to the main product, while product.product is mainly used for the variants. If you don't use variants, you'll have as many records on product.template as you have on product.product.*

Example: you sell yellow, red and blue T-shirts. You could configure two ways:

- As 3 different products: you create 3 records on the model product.template then you'll automatically have the 3 products on product.product: Red T-Shirt, Yellow T-Shirt, Blue T-Shirt

product.product	
ID	Name
1	Red T-Shirt
2	Yellow T-Shirt
3	Blue T-Shirt

- As 1 product (T-shirt) in the product.template with 3 variants (yellow, red and blue) that means 1 record on product.template and 3 records on product.product

product.template	
ID	Name
1	T-shirt

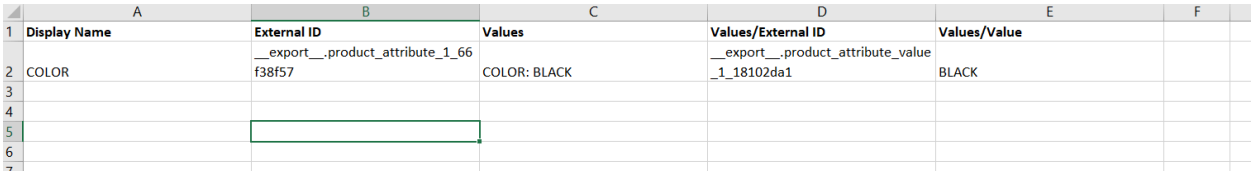
product.product		
ID	Product	Name
1	1	Red T-shirt
2	1	Yellow T-shirt
3	1	Blue T-shirt

The two models are synchronized in terms of data except for some fields : if you update a field on the product.template model, its value on the product.product model will automatically be updated. The two models are not synchronized in terms of view: if you edit the view on product.template, it will not update the view on product.product. See this video here for more information.

Products and their attributes

If you need to mass import available attributes to a product, you can do so by collecting the respective external IDs. Start with the attributes:

- If you have not yet imported any attributes, you can mass import including the external ID. These IDs are the attribute external IDs.
- If you have already imported attributes manually, navigate to the attribute table (Configuration > Attribute). In your product Attribute table select the attribute that you wish to upload to a product and export the following columns (Your file will be: product.attribute):
    - Display Name, External ID, Values, Values/External ID, Values/Value



	A	B	C	D	E	F
1	**Display Name**	**External ID**	**Values**	**Values/External ID**	**Values/Value**	
2	COLOR	__export__.product_attribute_1_66f38f57	COLOR: BLACK	__export__.product_attribute_value_1_18102da1	BLACK	
3						
4						
5						
6						

Next, you will collect the external IDs for the products that need the attributes added to their list:

- Navigate to the Product table (Product > Product) and in list view select and export the product for which you wish to add an attribute and its values. You only need to export the external ID (Your file will be: product.template).

On the product.template file add the following columns next to external ID:

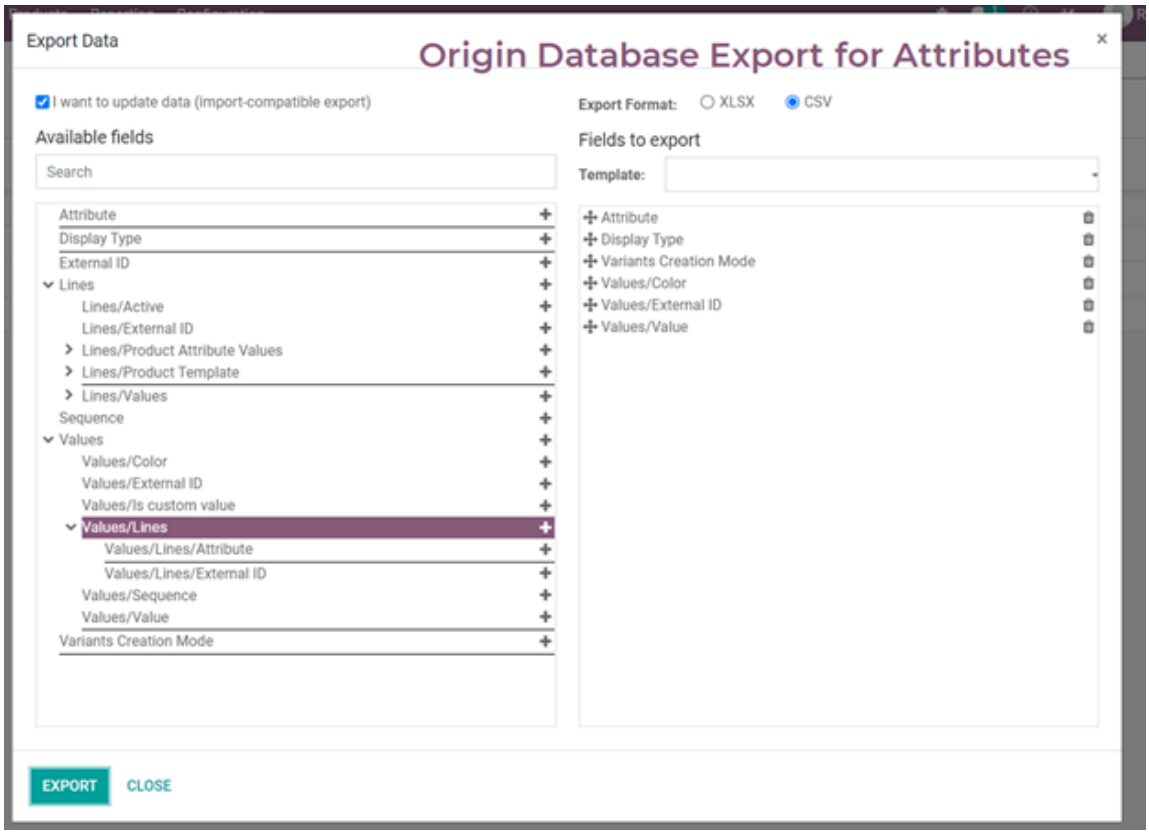
- Attribute_Line_Ids/Attribute_ID/ID, Attribute_Line_IDs/Value_IDs/ID

In the “Attribute_Line_Ids/Attribute_ID/ID” column on the product.template add the data from the “External ID” column in product.attribute. In the “Attribute_Line_IDs/Value_IDs/ID” column on product.template add the data from the “Values/External ID” column. If you have multiple values, they must be concatenated.

Import the product.template file into the Product list view.

Product Attribute Export & Import

This section describes how to export/import attributes with no related products associated. From the attribute page, select the attributes you'll export, and under "Action" click Export. Select the import-compatible export. The fields in the screenshot can be used as a template to modify and re-import to update existing attributes.

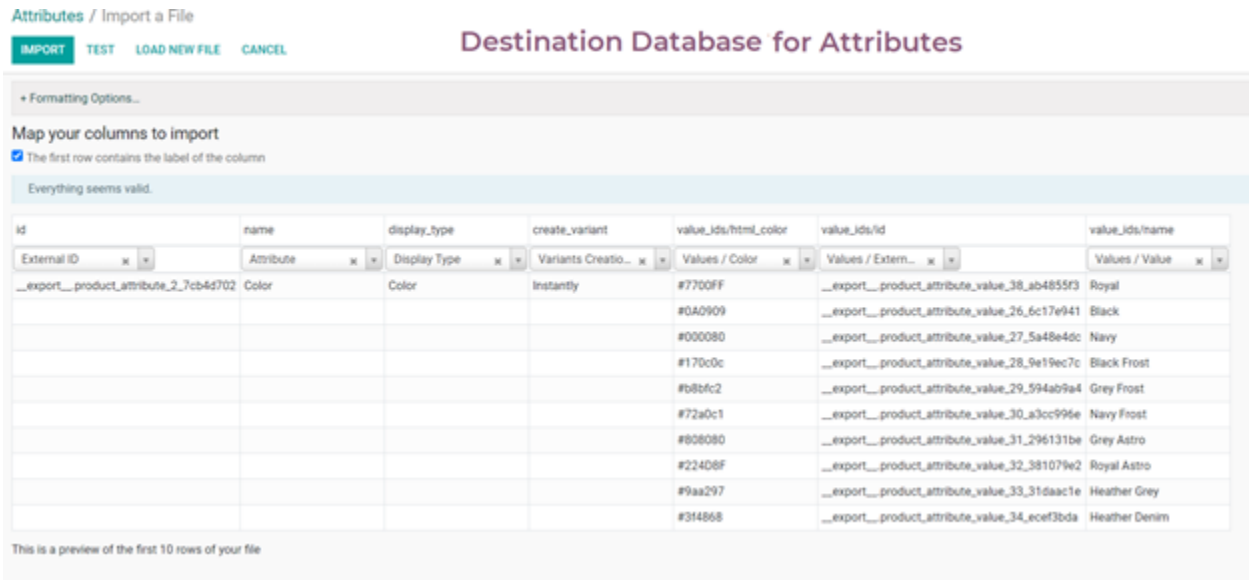


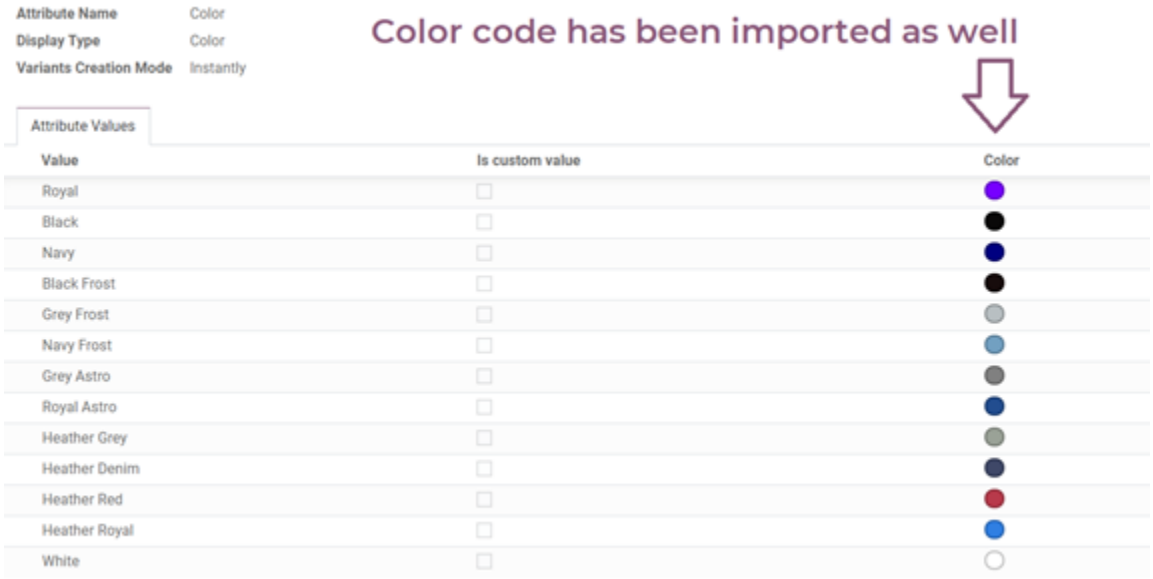
In the downloaded excel spreadsheet, we see the Attribute name "Color, Size, and TCC115", as well as their display type. Note that "Color" has the html code column as well.

Attribute Export: Each value name will have an external ID. Noticecolors can have their # exported as well for reimport. Attributes and values could also be added this sheet

id	name	display_type	create_variant	value_ids/html_c	value_ids/id	value_ids/name
__export__.product_	Color	Color	Instantly	#7700FF	product_attribute_value-jhc-00100	Royal
				#0A0909	product_attribute_value-jhc-00101	Black
				#000080	product_attribute_value-jhc-00102	Navy
				#170c0c	product_attribute_value-jhc-00103	Black Frost
				#b8bfc2	product_attribute_value-jhc-00104	Grey Frost
				#72a0c1	product_attribute_value-jhc-00105	Navy Frost
				#808080	product_attribute_value-jhc-00106	Grey Astro
				#224D8F	product_attribute_value-jhc-00107	Royal Astro
				#9aa297	product_attribute_value-jhc-00108	Heather Grey
				#3f4868	product_attribute_value-jhc-00109	Heather Denim
				#B83A4B	product_attribute_value-jhc-00110	Heather Red
				#307FE2	product_attribute_value-jhc-00111	Heather Royal
				#FFFFFF	product_attribute_value-jhc-00112	White
__export__.product_	Size	Radio	Instantly		product_attribute_value-jhc-00113	XS
					product_attribute_value-jhc-00114	S
					product_attribute_value-jhc-00115	M
					product_attribute_value-jhc-00116	L
					product_attribute_value-jhc-00117	XL
					product_attribute_value-jhc-00118	2XL
					product_attribute_value-jhc-00119	3XL
__export__.product_	TCC115	Radio	Instantly		product_attribute_value-jhc-00120	Captain
					product_attribute_value-jhc-00121	Firstmate
					product_attribute_value-jhc-00122	Mr.
					product_attribute_value-jhc-00123	Mrs.
					product_attribute_value-jhc-00124	PirateCaptain
					product_attribute_value-jhc-00125	PirateFirstmate

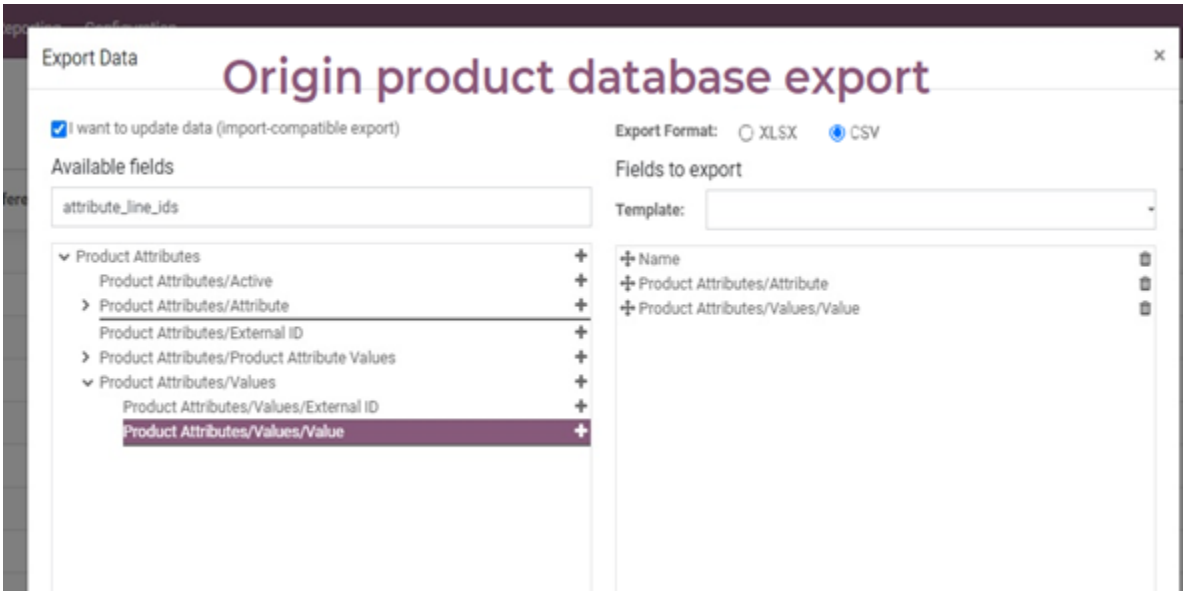
This template will natively map back into Odoo, and does not require any additional fields to import back into Odoo.



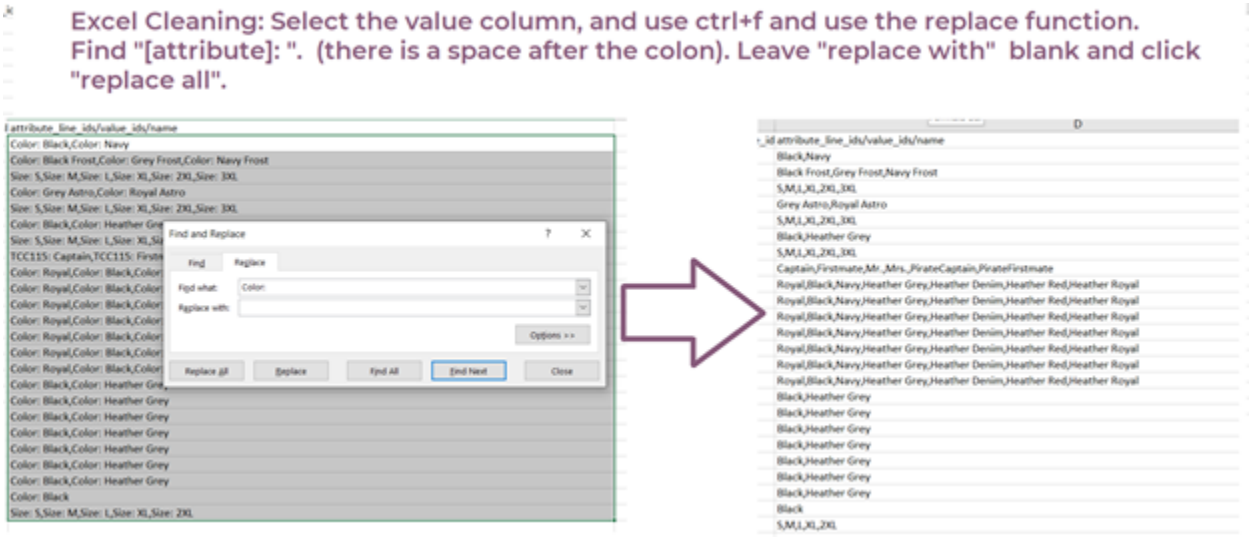


Product Variant Export & Import

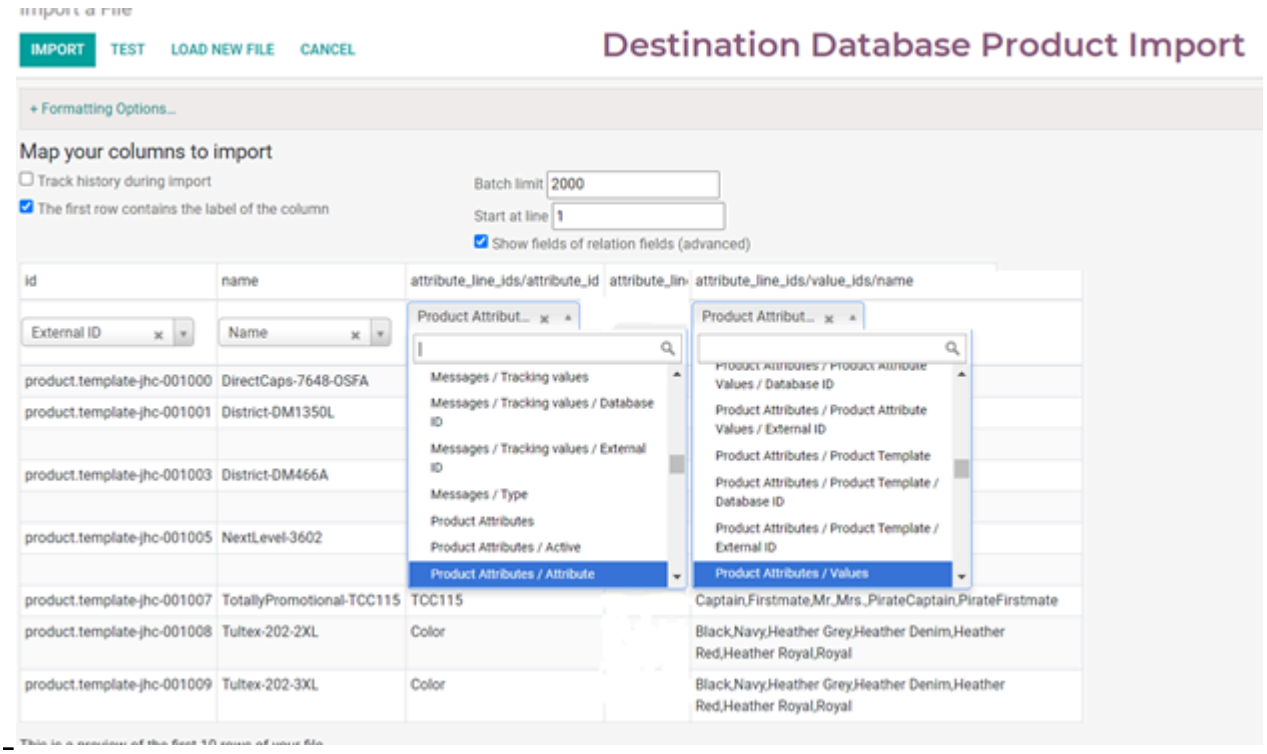
This section describes how to export and import a product with their attributes in a way that will create variants (product.product) inside a product.template. This is useful when exporting products and their variants from a test database into a production database. Begin by exporting products from the product page. Check the box for “import-compatible export” and then select the three columns below. Also change the export format to CSV for this.



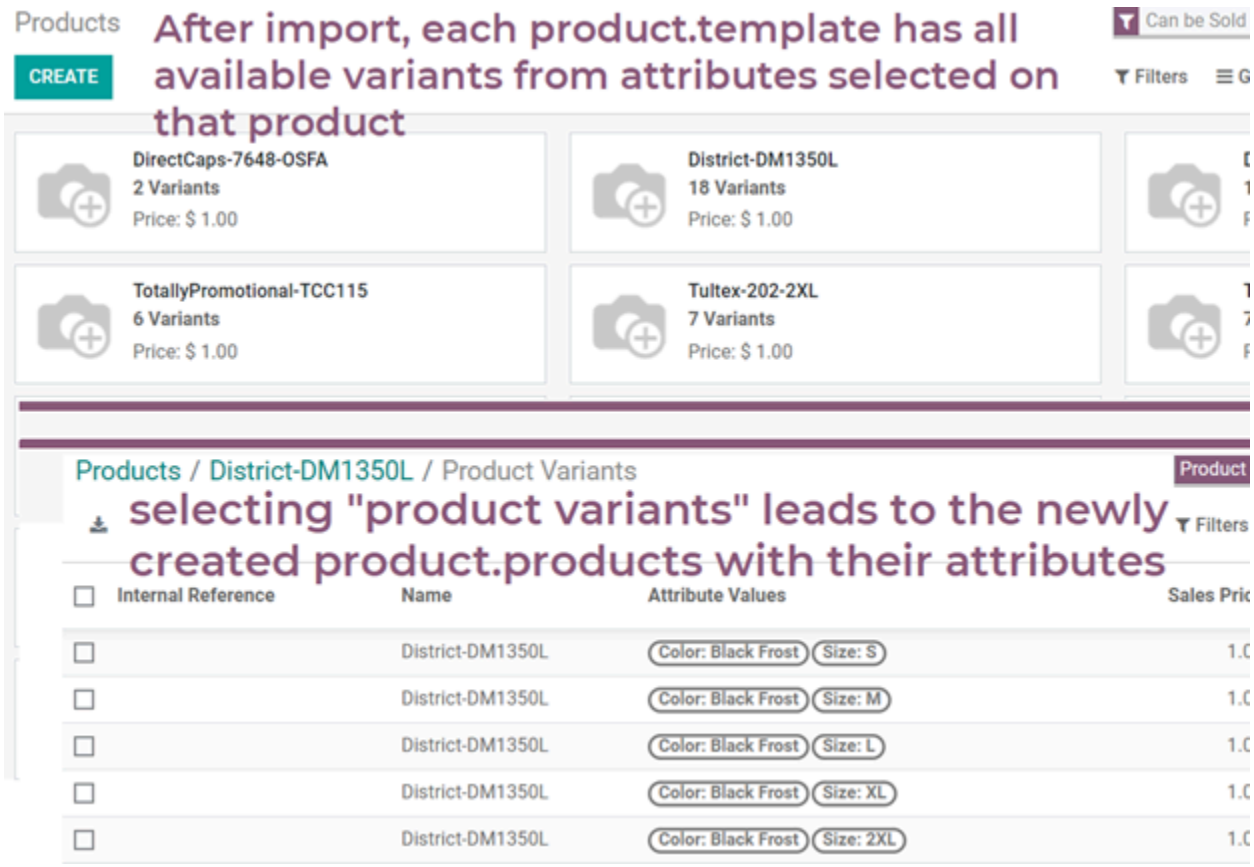
Following export, the delineation of “[attribute name]: ”, will need to be deleted before each value. Using the find&replace function in excel will make this simple. Find the “[attribute name]: “, for each attribute you have and replace them all with nothing. Make sure to do it correctly, as you won’t be able to import if you deleted a character off a value or left a space attached that’s not found on the variant name.



When importing this template, select “Product Attribute/ Attribute” for the attribute name column, and “Product Attribute/ Values” for the values.



This has imported the product.template and also created a product.product for each available combination of variants within the product.template.



Booleans

The boolean field (checkbox) is a binary value, that means that it's either checked or not. When you import such a field, write "TRUE" if the field should be checked, otherwise write "FALSE" (whether using small or upper letters). The values "0" (false) and "1" (true) also work.

One to Many (o2m) fields

A o2m field is a table with data (ex: the sale order lines, the journal items in a journal entry, etc.). All the data contained in a o2m field (ex: the unit price and the ordered quantities on the sale order line) correspond to a field in this table and should be filled in a specific cell in the Excel file. The column title of this field is built as follows : "technical name of the o2m field / technical name of the field included on the o2m field". For instance, if we want to import the unit price on the sale order line, the column title should be "order_line/price_unit"

As you create a new record in the related model, you'll need to include an external id for each line of the o2m field. For instance, if you want to import order lines on the sale order, you'll have to add a column whose title would be "order_line/id"

Computed fields

Odoo uses a piece of Python code to calculate the value in a field. It could be a calculated string, number, boolean, etc. Examples: product display name, invoice total, inventory level.

We never import a computed field. For instance, we don't import the total amount of the sales order line, as it's automatically computed based on the imported unit price, quantity, discount, etc.

Opening Trial Balance / Opening Inventory

For instructions on importing opening trial balance and inventory, see the Accounting Go-Live Checklist.