



SYMSON DATA FORMAT

12

Description

This document describes the data required for enabling the solutions offered by SYMSON. There can be different requirements depending on the chosen solution.

All solutions require a [Products](#) data import. Solutions including Machine Learning - Margin Optimization, require additionally an [Order Lines](#) data import.

Products Data

The table below lists all the fields supported in SYMSON for [Products](#) data import. Not all fields are required and depending on the solution, the list of required fields varies.

The data should be supplied in one csv file using a consistent field delimiter, and as columns the fields listed below. Check the [csv sample dataset](#) for an example and general technical specifications at the end of the document.

The data can also be provided using the SYMSON APIs. For the technical specifications, visit: <https://data.symson.com/swagger/index.html>

Field	Type	Description
ProductId (Required)	String	Unique product identifier.
EAN	String	European Article Number – used for tracking competitor prices.
Name	String	Name of the product.
Description	String	Short product description.
Brand	String	Brand of the product.
Supplier	String	Name of the product supplier.



ProductMainGroup	String	Main category used for classifying products in a hierarchy.
ProductSubGroup	String	A sub-category used for classifying products in a hierarchy.
ProductSubSubGroup	String	A sub-sub-category used for classifying products in a hierarchy.
ProductAttributeA	String	Additional product information/characteristic that is not supported by one of the previous fields.
ProductAttributeB	String	Same as above.
ProductAttributeC	String	Same as above.
ProductTotalCost (Required)	Float	<p>Cost of the product. This is the value used to determine the margin (as the difference between ListPrice and ProductTotalCost).</p> <p>For a detailed specification of the costs, use the following fields: ProductPurchaseCost, ProductShippingCost, ProductOtherCost, FeePercentage.</p> <p>These are optional, but if supplied, the ProductTotalCost needs to equate to:</p> $\text{ProductTotalCost} = \text{ProductPurchaseCost} + \text{ProductShippingCost} + \text{ProductOtherCost} + \text{ListPrice} * (1 + \text{VatPercentage}) * \text{FeePercentage}$
ProductPurchaseCost	Float	Used for a detailed specification of the costs. Purchase cost of the product.
ProductShippingCost	Float	Used for a detailed specification of the costs. Shipping costs – from the SYMSON user to their customers.
ProductOtherCost	Float	Used for a detailed specification of the costs. Other costs that are not purchase or shipping costs – as explained above.
FeePercentage	Float	Used for a detailed specification of the costs. Platform/market-place fee percentage or any other fee percentage (example: royalty rates) that needs to be paid as a percentage of the price of the product(including VAT). If there are multiple fees, sum these together.
Stock	Float	Number of products in stock.
VatPercentage	Float	The vat percentage used for selling the product.
ListPrice (Required)	Float	Current price of the product – excluding VAT.
ProductSupplierListPrice	Float	List price recommended by the Supplier – excluding VAT. This is not the purchase price.
ProductRRP (Recommended Retail Price)	Float	Recommended Retail Price – excluding VAT.
LowerBoundPrice	Float	Minimum price allowed for the product – excluding VAT.



UpperBoundPrice	Float	Maximum price allowed for the product– excluding VAT
RegionId (Required)	String	Name of the region the product belongs to. If the product belongs to(is sold into) more than one region, add one entry(row in the data) for each region. The region can be a country(example: NL) or a platform/market-place(example: Bol) or a combination(example: Bol-NL).

The following table indicates the required fields based on the chosen SYMSON solution.
The solutions are grouped based on their requirements into:

1. Pricing Management
 - Cost Based & Business Rules
 - Machine Learning – Margin Optimization
 - Competitor Pricing
2. Scenarios With Machine Learning – Margin Optimization

	Pricing Management			Scenarios With Machine Learning- Margin Optimization
	Cost Based & Business Rules	Machine Learning - Margin Optimization	Competitor Pricing(EAN)**	
ProductId	Required	Required	Required	Required
EAN			Required	
Name				
Description				
Brand				
Supplier				
ProductMainGroup				
ProductSubGroup				
ProductSubSubGroup				
ProductAttributeA				



ProductAttributeB				
ProductAttributeC				
ProductTotalCost	Required	Required	Required	Required
ProductPurchaseCost				
ProductShippingCost				
ProductOtherCost				
FeePercentage				
Stock	Required*			
VatPercentage	Required*		Required	
ListPrice	Required	Required	Required	Required
ProductSupplierListPrice	Required*			
ProductRRP (Recommended Retail Price)	Required*			
LowerBoundPrice	Required*			
UpperBoundPrice	Required*			
RegionId	Required	Required	Required	Required

*These fields become required depending on the specific pricing strategy blocks used.

**For Competitor Pricing/Scraping not EAN-based please contact SYMSON Support to discuss specific requirements.

Please note that while the fields ProductMainGroup, ProductSubGroup, ProductSubSubGroup, ProductAttributeA, ProductAttributeB, ProductAttributeC and Brand are optional fields, it is highly recommended to import (some of) them into SYMSON. These fields can be used to build the product hierarchy that will facilitate the assigning of product groups to specific pricing strategies and configuring product group characteristics such as margins, discounts, key competitors.

Order Lines Data

For SYMSON solutions involving Machine Learning – Margin Optimization, [Order Lines](#) data is also required. The table below lists all the fields supported in SYMSON for [Order Lines](#) data import. Not all fields are required.



The data should be supplied in one csv file using a consistent field delimiter, and as columns the fields listed below. Check the [csv sample dataset](#) for an example and general technical specifications at the end of the document.

The data can also be split and supplied in four separate csv files: [Invoices](#), [Regions](#), [Customers](#), [Order Lines](#). The last column of the table indicates which field belongs to which file. Please note that some of the *Id* fields need to be supplied in two files.

The data can also be provided using the SYMSON APIs. For the technical specifications, visit: <https://data.symson.com/swagger/index.html>

Field	Type	Status	Description	File
InvoiceId	String	Required		Invoices
				Order Lines
InvoiceCreateDate	Date/DateTime	Required	The date when the invoice was created. This date will be used for filtering the order-lines, if optimization is performed on specific periods and time intervals.	Invoices
InvoiceAttributeA	String			Invoices
InvoiceAttributeB	String			Invoices
InvoiceAttributeC	String			Invoices
RegionId	String	Required	The identifier for the region the order-line belongs to. This allows SYMSON to separate the order-lines such that currency or market-specific business strategies don't affect the optimization. The region can be a country(example: NL) or a platform/market-place(example: Bol) or a combination(example: Bol-NL).	Regions
				Invoices
RegionName	String		The name of the region (example: Netherlands).	Regions
RegionCurrency	String			Regions



RegionAttributeA	String			Regions
RegionAttributeB	String			Regions
RegionAttributeC	String			Regions
CustomerId	String	Required		Customers
				Invoices
CustomerName	String			Customers
CustomerDescription	String			Customers
CustomerAttributeA	String			Customers
CustomerAttributeB	String			Customers
CustomerAttributeC	String			Customers
ProductId	String	Required		Order Lines
OrderLineId	String	Required		Order Lines
OrderLineCreatedDate	Date/DateTime		Used in case an order-line has been added on a date different than the date when the invoice was created.	Order Lines
Quantity	Float	Required	Can be negative in case of an return, but not 0.	Order Lines
PricePerUnitVat	Float		The price – including VAT of one unit sold.	Order Lines
PricePerUnit	Float	Required	The price – excluding VAT of one unit sold. It should equate to: $PricePerUnit = TotalCostPerUnit + MarginPerUnit$.	Order Lines
MarginPerUnit	Float	Required	The margin made per one unit sold.	Order Lines
TotalCostPerUnit		Required	Total cost of one unit sold. For a detailed specification of the costs, use the following fields: PurchaseCostPerUnit, ShippingCostPerUnit, OtherCostPerUnit, FeePercentage. These are optional, but if supplied, the TotalCostPerUnit needs to equate to:	Order Lines



			$TotalCostPerUnit = PurchaseCostPerUnit + ShippingCostPerUnit + OtherCostPerUnit + PricePerUnit * (1 + VatPercentage) * FeePercentage$	
PurchaseCostPerUnit	Float		Used for a detailed specification of the costs. Purchase cost per one unit.	Order Lines
ShippingCostPerUnit	Float		Used for a detailed specification of the costs. Shipping costs – from the SYMSON user to their customers per one unit. If the shipping costs are per invoice, split these cost proportionally to each unit sold.	Order Lines
OtherCostPerUnit	Float		Used for a detailed specification of the costs. Other costs per one unit, that are not purchase or shipping costs – as explained above.	Order Lines
FeePercentage	Float		Used for a detailed specification of the costs. Platform/market-place fee percentage or any other fee percentage (example: royalty rates) that needs to be paid as a percentage of the price of the product (including VAT). If there are multiple fees, sum these together.	Order Lines
VatPercentage	Float			Order Lines
OrderLineAttributeA	Float			Order Lines
OrderLineAttributeB	Float			Order Lines
OrderLineAttributeC	Float			Order Lines

General Technical Considerations

1. Data should be supplied in csv format (one file for [Products](#) data and one file for [Order Lines](#), if applicable – or four files, for the split option) with columns the fields listed above and a consistent field delimiter (same delimiter for each column and row)
2. We recommend string fields to be quoted (with double quotes ex: “this is a string field”)



3. All string fields can have a maximum length of 100 characters, except for *Name* and *Description* fields which can have a maximum length of 200 characters
4. All numeric fields must have the decimal point. No currency signs. For example:
 - 0.4 is correct
 - 0,4 is not correct
 - 4€ is not correct
5. Date fields may also include time and must follow the [ISO8601](#) format:
 - 2020-02-25
 - 2020-03-23T14:11:50Z