

## 4.2.2. Item Information

This data piece is the item master data. It describes all characteristics of items required to plan your inventory in different cases, such as:

- [planning demand at a category level](#);
- [accounting for constraints](#);
- [multi-echelon planning](#);
- [planning intersite transfers](#);
- [planning products with a shelf life](#).

Despite all the above cases, basically, the primary goal of this data piece is to provide Streamline with the current on-hand quantity for every [planning item](#). A planning item is defined by **Item code** or by pair **Item code, Location**, if you plan your inventory in different locations. Thus, the required data types for this data piece are shown in the table below.

Data name	Description	Datatype
<b>Item code</b>	The item identifier.	String
<b>Location</b>	The code of location where <b>Item code</b> is stored.	String
<b>On hand</b>	The current on-hand of <b>Item code</b> in <b>Location</b> .	Integer

The unit of measure the **On hand** provided in, is taken as the base unit of measure in Streamline.

## Demand Planning

Streamline allows you to plan your demand at a category level. It means that you are able to set specific [forecasting options](#) and [make manual forecast overrides](#) at a particular level of the category tree. To be able to do this, you should provide Streamline with the data types shown in the table below.

Data name	Description	Datatype	Is not given	
			Default	Provided
<b>Item category, Item sub-category</b>	Used to forecast by item categories.	String	Category with empty name	NULL or empty string
<b>Location category, Location sub-category</b>	Used to forecast by location categories.			

Empty categories at the end of the hierarchy are perfectly possible:  $Category1 > Category2 > \emptyset > \emptyset$  means that the item will be put into the  $Category2$ . The symbol  $\emptyset$  means an empty category.

Empty categories anywhere else in the hierarchy are also allowed but discouraged. For example,  $Category1 > \emptyset > Category3$ . In this case Streamline creates a subcategory with an empty name.

You can import as many categories as you need.

In addition to the **Item code** and **Location**, you can import data types displayed in the table below.

Data name	Description	Datatype	Is not given	
			Default	Provided
<b>Item description</b>	A description of the item. It is used to better understand what a particular item code means.	String	Empty string	NULL or empty string
<b>Location description</b>	A description of the location. It is used to better understand what a particular location code means.			
<b>Info field</b>	Any additional characteristic of the item (e.g., color, size, texture). You can import an unlimited number of such fields.			

## Inventory Planning

If your data source contains the relevant lead time and order cycle, Streamline gives you the ability to import them through data types indicated in the table below.

Data name	Description	Given in	Datatype	Default
<b>Lead time</b>	If the planning item is sourced from a supplier, this is the average supplier lead time; if it is sourced from a <a href="#">distribution center</a> (DC), this is the average lead time to deliver ordered items from the DC to the <a href="#">Location</a> . These lead times are interpreted by Streamline as the interval of time between purchase/transfer order placement and its receipt.	Days	Integer	30
<b>Order cycle</b>	If the planning item is sourced from a supplier, this is the frequency you order from this supplier; if it is sourced from a DC, it is the frequency you replenish from the DC.	Data aggregation periods (weeks or month), days, or the <b>Lead times</b> .		1 data aggregation period

By default, Streamline considers that a planning item is sourced from a supplier. To set it to be supplied from a DC, the [DC name](#) data type is used. Be aware of that, when [providing Lead time](#) and **Order cycle** for each planning item.

To get accurate inventory replenishment plans along with the **Lead time**, **Order cycle**, and [Transaction data](#), **we strongly recommend** that you provide Streamline with the [Orders-to-receive information](#) and [Orders-to-ship information](#).

The accuracy of inventory reports can be improved, if you additionally provide the **Lead time variance** (see table below).

Data name	Description	Given in	Datatype
<b>Lead time variance</b>	The variance of the <b>Lead time</b> .	<b>Lead time</b> units	Float

To optimize the inventory, Streamline uses a lot of inputs such as **Last on hand**, **Lead time**, **Order cycle**, and other. [Safety stock](#) can be also considered as an input because it is used to determine the optimal ordering plan.

There two basic methods that Streamline uses to calculate safety stock. The first one is a [well-known method](#) based on service level. Streamline [implements it](#) and allows you to import service level for each planning item from a database. Another method is based on the [future demand](#). Streamline allows you to set up the number of future periods of which demand is taken as the safety stock for a planning item.

A description of these data types is given in the table below.

Data name	Description	Given in	Datatype
<b>Service level</b>	It is the probability of product availability in stock. It reflects the ability to sell the product to a customer.	Fractional number in the semi-open interval [0, 1)	Float
<b># of periods for safety stock</b>	The number of future periods of which demand is used as the safety stock.	Data aggregation periods	

Streamline allows importing settings on how each item should be planned right from the data source. The table below shows which settings are available to set up.

Setting	Description	Given in	Datatype
<b>Inactive</b>	Automatically sets the <a href="#">inactive</a> model type to the imported item.	You can provide <b>1</b> or <b>'true'</b> , or <b>'yes'</b> to put this setting in action, or <b>0</b> , <b>'false'</b> , or <b>'no'</b> - to ignore it.	Integer or String
<b>Termination</b>	Automatically sets the <a href="#">Auto &amp; termination</a> model type to the imported item.		
<b>Purchase for BOM</b>	By default, if an item has a BOM, Streamline automatically treats this item as a manufactured one and builds a material procurement plan for its components. But, in the case, you want to purchase it, not manufacture, this setting tells Streamline to treat it as a regular item that is purchased. It also sets the <a href="#">material procurement</a> setting to the <b>Purchase</b> for the imported item.		

## Data Types for Inventory KPIs

Streamline can calculate key inventory indicators (KPI) such as expected stockout and overstock values, gross margin, and other [metrics](#). To enable the calculations, you should provide one of the data types shown in the table below.

Data name	Description	Datatype
<b>Inventory value/unit</b>	The balance value of one unit of the item in stock. It should be given in the <a href="#">base currency</a> . It is preferable than the <b>Item purchase price</b> described below.	Float
<b>Purchase price/unit</b>	The last purchase price of the item in the supplier's currency.	
<b>Sales price/unit</b>	The current sales price of the item.	

The **Purchase price/unit** is also used to calculate the [value](#) of purchase orders' lines recommended by Streamline. You need to import either **Sales price/unit** or **Transaction revenue** for some of the KPIs (for example, the annual revenue, revenue next year, etc.).

**Sales price/unit** given through the *Item information* data piece has a higher priority than the last sales price determined from the [Transactions](#).

If [supplier's currency](#) is not given, Streamline implies that **Item purchase price** is given in the [base](#)

[currency](#).

## Supplier's Information

Streamline allows you to import supplier information shown in the table below.

Data name	Description	Datatype
<b>Supplier code</b>	The identifier of the supplier.	String
<b>Supplier's currency</b>	The currency in which purchase orders are placed.	
<b>Supplier's item code</b>	Item code in supplier's stock-list that corresponds to your <b>Item code</b> . It allows Streamline to <a href="#">create</a> purchase orders in supplier's item codes.	

If **Item value/unit** is not given, **Item purchase price** and **Supplier's currency** are given, then [gross-margin](#) and [turn-earn index](#) will not be calculated.

## Accounting for Constraints in Optimization

Streamline allows you to account for constraints when optimizes inventory. There are two types of constraints, constraints on the item quantity that is ordered and constraints on the purchase order as a whole. Data types for the constraints are given in the table below.

Data name	Description	Datatype
<b>Display qty</b>	The minimum number of units a shelf to display. This parameter typically arises in the retail business. You can <a href="#">control</a> how this amount is used when <a href="#">final safety stock</a> is calculated.	Integer
<i>Constraints on item quantity ordered</i>		
<b>Min lot and Max lot</b>	Defines optional constraints on how few or how many of the planning item you can replenish from the supplier or DC with one order.	Integer
<b>Rounding</b>	This parameter rounds up the <a href="#">Net order quantity</a> calculated by Streamline. This allows Streamline to take into account how many items are included in a carton, allowing the application to suggest exactly the right amount of the item to order.	
<i>Constraints on purchase order</i>		
<b>Group ID</b>	Identifies a group of items on which minimum order constraints such as Group min/qty/weight/volume/cost are applied. If <b>Supplier code</b> is imported, it is automatically assigned to Group ID.	String
<b>Group min. qty</b>	The minimal quantity/weight/volume/cost for the item group required in order to replenishment/purchase order can be accepted.	Integer
<b>Group min. weight/volume/cost</b>		Float
<b>Weight/unit</b>		
<b>Volume/unit</b>	The volume of one unit of an item.	
<b>Container ID</b>	An identifier that indicates to which container family the Item code belongs.	String

Data name	Description	Datatype
<b>Container load weight/volume/qty</b>	Container characteristics such as maximal weight, volume, or quantity of items you can load into a container of the same Container ID.	Integer
<b>Max containers</b>	Defines how many containers of the same Container ID you can order at a time.	Integer

Constraints on the purchase order are tied to the [Supplier code](#). If no supplier code is given they are attached to an empty supplier. If a planning item is supplied from a DC and has **Supplier code** and some of these constraints given, Streamline neglects these constraints.

Sometimes **Min lot**, **Max lot**, and **Rounding** constraints depend on the source the item is supplied from. For example, these parameters may be different whether you replenish a planning item from a DC or a supplier.

If **Min lot** is greater than **Max lot**, Streamline neglects **Min lot** and uses **Max lot** in the optimization. Order constraints such as **Supplier's min. weight** or **Supplier's min. volume** should be given to Streamline along with the **Weight/unit** and **Volume/unit** relatively.

Constraints on a purchase order are incompatible with the [product shelf life](#) limitation. It means that Streamline's purchase order recommendations come from the given purchase order constraints, not the shelf life limitation if both are given.

2019/04/03 09:29 · admin

## Two-echelon Planning

Streamline allows performing [two-echelon planning](#).

If you have only one [distribution center](#) (DC), Streamline does not require you to provide any additional data, and you can set up all DC's options in the [DC settings](#). In this case, however, there is no ability to account for the case when a location is supplied by the supplier directly (skipping the DC).

If you have several DCs, you can import and set up them using the **Database connection**. In this case:

- you should provide additional information (described below);
- distribution center is always [enabled](#) and the **Location** control is disabled in the **Settings**;
- a location can be supplied by the supplier directly skipping the DCs;
- the options set in the [DC settings](#) are applied to all your DCs at once.

You can import any number of DCs and set up which locations are supplied by a particular DC on an item basis. It means you should set up a triple (**Location, DC name, Item code**). For example, the triple (*West, DC west, Dark chocolate*) means that *DC west* supplies *West* location with *Dark chocolate*.

There are two limitations on the DC-location relation:

- Two (or more) DCs can't supply the same item to the same location.
- A DC can't supply another DC, that is, only DC-to-location relations are allowed.

To set up the relations, the **DC name** data type should be returned with the [Item info query](#) (see table below). It indicates the name of the DC in the triple.

Data name	Description	Datatype
<b>DC name</b>	The name of the distribution center that supplies the <b>Item code</b> to the <b>Location</b>	String

Now, we describe the data types that the *Item info query* should return in order to Streamline set the relations properly.

As we explained previously, to set up a triple, the *Item info query* should return the following data columns: **Location**, **DC name**, and **Item code**. The table below shows an example of records that should be returned by the query in order to set up a DC-location-item relation.

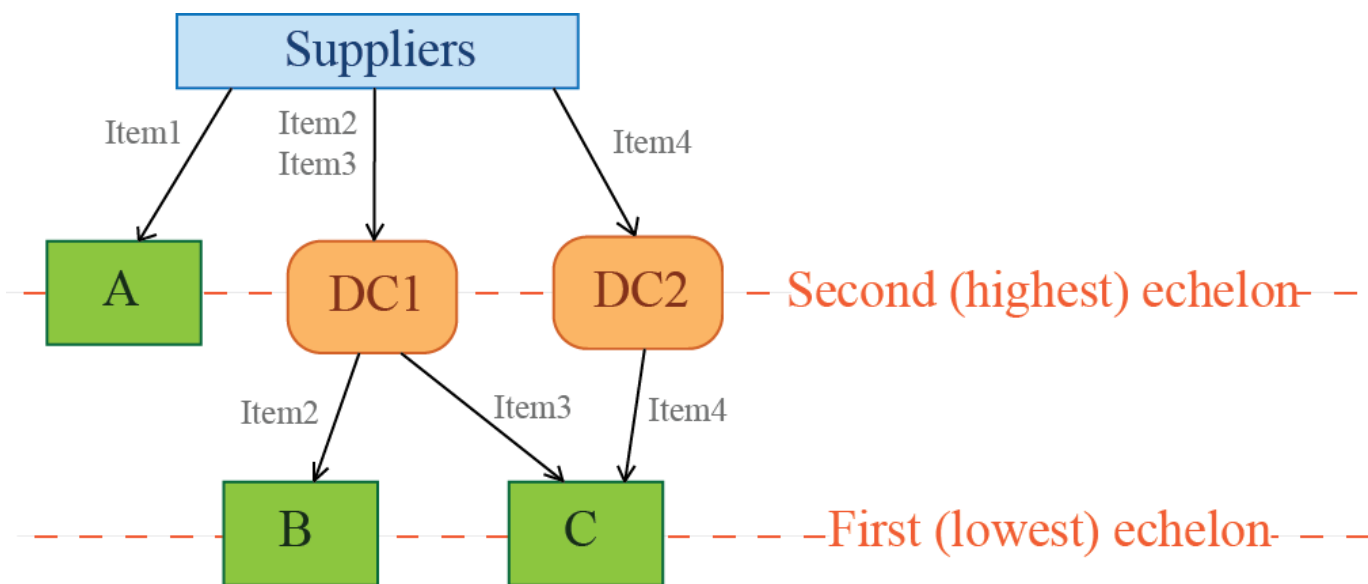
Location	DC name	Item code
A	DC1	Item1
DC1	NULL	Item1

As you see, we need two records to be returned for each DC-location-item relation. The first one links location A to DC1, meaning that *Item1* will be supplied by DC1 to location A. The second one declares DC1 as a location that stores *Item1*.

To set up a situation when an item is supplied to a location by the supplier directly (no DC involved), the query should return the record shown in the table below. The table shows example data.

Location	DC name	Item code
B	NULL	Item1

Let's consider an example shown in the figure below.



In this case, the query should return the data shown in the table below.

Location	DC name	Item code
A	NULL	Item1
B	DC1	Item2
C	DC1	Item3
C	DC2	Item4

Location	DC name	Item code
DC1	NULL	Item2
DC1	NULL	Item3
DC2	NULL	Item4

## Intersite Optimization

Streamline can [generate suggestions](#) on inventory transfers between your stores if there is an overstock at least at one of them. By default, Streamline spends this overstock to fulfill [current orders](#) going up from the smallest to the largest, maximizing the number of replenished stores.

Additionally, Streamline allows you to put constraints on this rule by introducing the regions where the transfers are allowed. This is done using a data type that should be set for each location belonging to a region (see table below).

Data name	Description	Datatype
<b>Transfer region</b>	The region the location belongs to	String

Locations belonging to different regions can't have transfers. At the same time, transfers between locations of the same region are allowed.

These constraints are optional, thus, **Transfer region** data type can have gaps, meaning that the locations do not take part in any intersite transfers.

## Planning Products with Shelf Life

Streamline allows you to [plan products having a limited shelf life](#). Product shelf life can be given in two units of measure (see table below).

Data name	Description	Given in	Datatype
<b>Shelf life, periods</b>	It is the desired time you want the product to be sold for.	Data aggregation periods	Float
<b>Shelf life, days</b>		Days	

The **Shelf life** parameter is used as a constraint in the inventory optimization. It is the maximal limit on the [current order quantity](#) derived from the given shelf-life period and generated demand forecasts.

[Next: Item Information](#)

[Download PDF](#)

From:

<https://gmdhsoftware.com/documentation-sl/> - **GMDH Streamline Docs**

Permanent link:

<https://gmdhsoftware.com/documentation-sl/imported-data-item-info>

Last update: **2023/05/19 15:53**

