Table of Contents

Overview ....................................................................................................................... 1
Setup .............................................................................................................................. 2
  Requirements ............................................................................................................. 2
  Installation .................................................................................................................. 2
  Setting Up The pdsalsobought Item ........................................................................ 3
Also Bought Products Batch List .................................................................................. 4
Technical Set-up ........................................................................................................... 5
Item Template ............................................................................................................... 7
JSON API ...................................................................................................................... 8
  AlsoBoughtProductsList_Load_Query ........................................................................ 9
    Request Parameters ............................................................................................... 9
    On Demand Columns ............................................................................................. 10
    Search Filter Columns ......................................................................................... 11
    Sort Filter Columns .............................................................................................. 11
    Example Request .................................................................................................. 12
    Example Response ............................................................................................... 12
  AlsoBoughtProduct_Update_Assigned ..................................................................... 13
    Request Parameters ............................................................................................... 13
    Example Request .................................................................................................. 14
    Example Response ............................................................................................... 14
  AlsoBoughtProducts_ProcessOrders .......................................................................... 15
    Request Parameters ............................................................................................... 15
    Example Request .................................................................................................. 15
    Example Response ............................................................................................... 15
XML Provisioning .......................................................................................................... 16
Uninstalling the Module ............................................................................................... 17
Overview

This module keeps a running count of products purchased together and provides a template layer to display the results at a product level. Each time an order is placed by a customer containing more than one item, all items in the order are added to each other’s also bought count, incrementing the count by one for each item. The also bought counts are tracked at the master product level. Items deleted from orders will decrement the also bought count for the deleted items by one on the remaining items.

New orders and/or order items added to existing orders in the admin or remotely via JSON API must use the Order Fulfillment Modules trigger to process the order and update the also bought products count.

1. Navigate to Order Processing > Orders in the Miva admin and edit an order to be processed.
2. Click Order Fulfillment Modules in the edit links section.
3. Check the Process checkbox for the Also Bought Products module.
4. Click the Process button to process the order.
Setup

Requirements

Miva Merchant: 10.00.00

Store User Interface: Miva Merchant CSSUI

Installation

1. Log into your Miva Merchant Admin
2. Navigate to Domain Settings > Modules
3. Click the Add Module to add a new Module
4. Click Upload and upload pdsalsobought.mvc
5. Press the Add button
6. Navigate to Modules > Add/Remove Modules
7. Find the Also Bought Products module in the modules list and click the Install button
8. The Also Bought Products module is now installed!
Setting Up The pdsalsobought Item

By default, this item should be created when the module is installed. In the case you do not see the item `pdsalsobought` in your items list, follow the steps below:

1. Navigate to **User Interface**
2. Click on the **Items** tabs
3. Click the **Add Item** button to add a new item
4. Set the code to `pdsalsobought`
5. Set the module to `pdsalsobought`
6. Click the **Add** button

You may now utilize the item on the pages you assign it to.
Also Bought Products Batch List

After the module is installed, a new **Also Bought Products** tab is available when editing a product in the Miva admin.

Clicking on the **Also Bought Products** tab link will take you to the also bought products batch list. The batch list allows you to view the current count and assignments for the also bought products. The filters and column displays function the same as the other batch lists found throughout the admin. Also bought products in the batch list may be assigned or unassigned by using the **ASSIGNED** column toggle. Unassigned also bought products will not be included in the product listing rendered by the `pdsalsobought` item.
Technical Set-up

To add also bought products to your page, you will need to do the following:

1. Navigate to User Interface
2. Click on the Items tab
3. Search for pdsalsobought and open the edit tab for that item.
4. Click on the Pages tab
5. Assign the item to the pages you would like to utilize the module.
   a. For example, if you want to assign the item to the product page, assign the item to the PROD page code.

Once you have assigned the item to the desired pages, the Also Bought Product List Layout template and options will appear on the assigned pages. The option settings control the custom fields, image types, sorting options, pagination, and image dimensions that are rendered with the template.

The Default Sort Method defaults to the also bought product count in descending order for the current product. For example, the most frequently bought product will be sorted first in the list for the Default option.
Also Bought Product List Layout page section
Item Template

To render the Also Bought Product List Layout template, place the following code within a page template where the pdsalsobought item is assigned to the page:

```
<mvt:if expr="l.settings:pdsalsobought_product_count">
  <mvt:item name="pdsalsobought" />
</mvt:if>
```

The \texttt{l.settings:pdsalsobought\_product\_count} variable contains the number of also bought products assigned to the current product.
The following functions are available via the JSON API for the module.

Since these are module level functions, you’ll need to use the following format to whitelist the functions:

```
Module:pdsalsobought:<Function Name>
```

Replace `<Function Name>` with the function names listed below.
AlsoBoughtProductsList_Load_Query

This function is used to query also bought products for a given product. The product to query for may be specified using `Product_ID`, `Product_Code`, or `Product_SKU`.

Only one of the product identifiers is required.

Request Parameters

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store_Code</td>
<td>String</td>
<td>Store code to run function against.</td>
</tr>
<tr>
<td>Function</td>
<td>String</td>
<td>This will always be the value <code>Module</code>.</td>
</tr>
<tr>
<td>Module_Code</td>
<td>String</td>
<td>This will always be the value <code>pdsalsobought</code>.</td>
</tr>
<tr>
<td>Module_Function</td>
<td>String</td>
<td>This will always be the value <code>AlsoBoughtProductsList_Load_Query</code>.</td>
</tr>
<tr>
<td>Product_Id</td>
<td>Number</td>
<td>Product id of the product to query.</td>
</tr>
<tr>
<td>Product_Code</td>
<td>String</td>
<td>Product code of the product to query.</td>
</tr>
<tr>
<td>Product_Sku</td>
<td>String</td>
<td>Product sku of the product to query.</td>
</tr>
<tr>
<td>Count</td>
<td>Number</td>
<td>Count is used to limit how many also bought products you want returned per request</td>
</tr>
<tr>
<td>Offset</td>
<td>Number</td>
<td>Used in conjunction with the <code>Count</code> parameter for paginating results of large datasets. Default is 0, which is no offset.</td>
</tr>
<tr>
<td>Sort</td>
<td>String</td>
<td>Field to sort results by. Available Sorting fields are listed below. Use – prefix for a descending sort.</td>
</tr>
<tr>
<td>Filter</td>
<td>Object</td>
<td>Array of Objects containing search filters. See usage here: List Load Query Overview. Available Search Filter fields are listed below.</td>
</tr>
<tr>
<td>Assigned</td>
<td>Boolean</td>
<td>Filter product list results to only include assigned also bought products. Defaults to true if omitted.</td>
</tr>
</tbody>
</table>

9
<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unassigned</td>
<td>Boolean</td>
<td>Include unassigned also bought products in product list results. Defaults to false if omitted.</td>
</tr>
</tbody>
</table>

**On Demand Columns**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>descrip</td>
<td>Full HTML product description</td>
</tr>
<tr>
<td>catcount</td>
<td>Number of categories the product is assigned to</td>
</tr>
<tr>
<td>productinventorysettings</td>
<td>Product specific inventory settings</td>
</tr>
<tr>
<td>attributes</td>
<td>Includes attribute object in response</td>
</tr>
<tr>
<td>productimagedata</td>
<td>Product Image including Image Types and non-image types</td>
</tr>
<tr>
<td>categories</td>
<td>Product Category Assignments</td>
</tr>
<tr>
<td>productshippingrules</td>
<td>Includes product dimensions and restricted shipping methods (if any)</td>
</tr>
<tr>
<td>relatedproduct</td>
<td>Related product assignments</td>
</tr>
<tr>
<td>uris</td>
<td>Includes URI's object in response. Canonical URI is the default URL for the product.</td>
</tr>
<tr>
<td>CustomField_Values:*</td>
<td>See custom field usage here: <a href="https://docs.miva.com/json-api/list-load-query-overview">https://docs.miva.com/json-api/list-load-query-overview</a></td>
</tr>
<tr>
<td>product_inventory</td>
<td>Includes 2 new fields for inventory counts as well as if inventory is active</td>
</tr>
<tr>
<td>page_code</td>
<td>Includes page_code field associated with product in response. Page_Code is used for alternate Product page templates.</td>
</tr>
<tr>
<td>cancat_code</td>
<td>Includes cancat_code field associated with product in response. cancat_code is the canonical category code for the product.</td>
</tr>
</tbody>
</table>
## Search Filter Columns

<table>
<thead>
<tr>
<th>id</th>
<th>code</th>
<th>sku</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>thumbnail</td>
<td>image</td>
</tr>
<tr>
<td>price</td>
<td>cost</td>
<td>descrip</td>
</tr>
<tr>
<td>weight</td>
<td>taxable</td>
<td>active</td>
</tr>
<tr>
<td>page_title</td>
<td>dt_created</td>
<td>dt_updated</td>
</tr>
<tr>
<td>cancat_code</td>
<td>page_code</td>
<td>alsobought_count</td>
</tr>
<tr>
<td>category</td>
<td>Supports only IN and NOT_IN operators. Accepts a comma separated list of category codes. Limits results to products in the specified categories.</td>
<td></td>
</tr>
<tr>
<td>product_inventory</td>
<td>Searches the product's inventory level (not accounting for baskets)</td>
<td></td>
</tr>
</tbody>
</table>

## Sort Filter Columns

One of these values would be passed in the sort request parameter to have the results sorted by that column. Value may be preceded by - for a descending sort.

<table>
<thead>
<tr>
<th>id</th>
<th>code</th>
<th>sku</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>thumbnail</td>
<td>image</td>
</tr>
<tr>
<td>price</td>
<td>cost</td>
<td>descrip</td>
</tr>
<tr>
<td>weight</td>
<td>taxable</td>
<td>active</td>
</tr>
<tr>
<td>page_title</td>
<td>dt_created</td>
<td>dt_updated</td>
</tr>
<tr>
<td>cancat_code</td>
<td>page_code</td>
<td>alsobought_count</td>
</tr>
</tbody>
</table>
Example Request

```json
{
    "Store_Code": "PS",
    "Function": "Module",
    "Module_Code": "pdsalsobought",
    "Module_Function": "AlsoBoughtProductsList_Load_Query",
    "Product_Code": "prod1"
}
```

Example Response

```json
{
    "success": 1,
    "data": {
        "total_count": 1,
        "start_offset": 0,
        "data": [
            {
                "id": 158130,
                "code": "heavy_prod",
                "sku": "",
                "name": "Heavy Product",
                "thumbnail": "",
                "image": "",
                "price": 500.00,
                "formatted_price": "$500.00",
                "cost": 0.00,
                "formatted_cost": "$0.00",
                "weight": 1350.00,
                "taxable": true,
                "active": true,
                "page_title": "",
                "dt_created": 1508189047,
                "dt_updated": 1618693596,
                "page_code": "ALT_P_TEST",
                "cancat_code": "",
                "product_inventory_active": true,
                "product_inventory": 100,
                "assigned": true,
                "alsobought_count": 12
            }
        ]
    }
}
```
**AlsoBoughtProduct_Update_Assigned**

This function allows you to assign or unassign an also bought product.

The product to target may be identified using `Product_ID`, `Product_Code`, or `Product_SKU`.

The also bought product to target may be identified using `AlsoBoughtProduct_ID`, `AlsoBoughtProduct_Code`, or `AlsoBoughtProduct_SKU`.

Only one of the product identifiers and one of the also bought product identifiers are required.

### Request Parameters

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store_Code</td>
<td>String</td>
<td>Store code to run function against.</td>
</tr>
<tr>
<td>Function</td>
<td>String</td>
<td>This will always be the value <code>Module</code>.</td>
</tr>
<tr>
<td>Module_Code</td>
<td>String</td>
<td>This will always be the value <code>pdsalsobought</code>.</td>
</tr>
<tr>
<td>Module_Function</td>
<td>String</td>
<td>This will always be the value <code>AlsoBoughtProduct_Update_Assigned</code>.</td>
</tr>
<tr>
<td>Product_ID</td>
<td>Number</td>
<td>Target product id.</td>
</tr>
<tr>
<td>Product_Code</td>
<td>String</td>
<td>Target product code.</td>
</tr>
<tr>
<td>Product_SKU</td>
<td>String</td>
<td>Target product sku.</td>
</tr>
<tr>
<td>AlsoBoughtProduct_ID</td>
<td>Number</td>
<td>Target also bought product id.</td>
</tr>
<tr>
<td>AlsoBoughtProduct_Code</td>
<td>String</td>
<td>Target also bought product code.</td>
</tr>
<tr>
<td>AlsoBoughtProduct_SKU</td>
<td>String</td>
<td>Target also bought product sku.</td>
</tr>
<tr>
<td>Assigned</td>
<td>Boolean</td>
<td>Pass <code>true</code> to assign also bought product. Pass <code>false</code> to unassign also bought product.</td>
</tr>
</tbody>
</table>
Example Request

```json
{
    "Store_Code": "PS",
    "Function": "Module",
    "Module_Code": "pdsalsoought",
    "Module_Function": "AlsoBoughtProduct_Update_Assigned",
    "Product_Code": "prod1",
    "AlsoBoughtProduct_Code": "heavy_prod",
    "Assigned": true
}
```

Example Response

```json
{
    "success": 1
}
```
AlsoBoughtProducts_ProcessOrders

This function is used to update the also bought counts for products by processing the order items from a list of orders.

**Request Parameters**

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store_Code</td>
<td>String</td>
<td>Store code to run function against.</td>
</tr>
<tr>
<td>Function</td>
<td>String</td>
<td>This will always be the value <em>Module</em>.</td>
</tr>
<tr>
<td>Module_Code</td>
<td>String</td>
<td>This will always be the value <em>pdsalsobought</em>.</td>
</tr>
<tr>
<td>Module_Function</td>
<td>String</td>
<td>This will always be the value <em>AlsoBoughtProducts_ProcessOrders</em>.</td>
</tr>
<tr>
<td>Order_IDs</td>
<td>Array</td>
<td>Array of order ids to be processed.</td>
</tr>
</tbody>
</table>

**Example Request**

```json
{
  "Store_Code": "PS",
  "Function": "Module",
  "Module_Code": "pdsalsobought",
  "Module_Function": "AlsoBoughtProducts_ProcessOrders",
  "Order_IDs": [
    97268,
    97267
  ]
}
```

**Example Response**

```json
{
  "success": 1,
  "processed": 2
}
```
XML Provisioning

The following XML Provisioning tags are available for use with Miva's Data Management import with the Import Data and Settings in XML Provisioning Format option.

```xml
<Module code="pdsalsobought" feature="fulfill">
  <!-- Unassign an also bought product from a product -->
  <AlsoBoughtProduct_Unassign product_code="prod1" alsoboughtproduct_code="prod2" />

  <!-- Assign an also bought product to a product -->
  <AlsoBoughtProduct_Assign product_code="prod1" alsoboughtproduct_code="prod2" />

  <!-- Process orders to update the also bought counts -->
  <AlsoBoughtProducts_ProcessOrder order_id="97265" />
  <AlsoBoughtProducts_ProcessOrder order_id="98124" />
</Module>
```
Uninstalling the Module

To uninstall the module, you will need to make sure that the `pdsalsobought` item is fully removed.

Once the item is removed:

1. Navigate to **Modules > Add/Remove Modules**
2. Find the **Also Bought Products** module
3. Click the `…` button and click **Uninstall**

The module will remove all of the also bought products data once uninstalled.