

Dev note

- [Dev notes](#)
- [Magento dev tricks and tips](#)
- [PHP files and XML files to edit](#)
- [Coding guidelines](#)
- [Dev bulletin 11/15](#)
- [Coding Suggestions](#)
- [Frans Extensions](#)

Dev notes

8/15/24: Latest js code for usps address validation:

Library: `app/code/Frans/CheckoutViews/view/frontend/web/js/addrval.js`

Using the above library: `app/code/Frans/CheckoutViews/view/frontend/web/js/view/shipping.js`

8/14/24:

`app/code/Frans/MultiShip/view/adminhtml/web/js/quoteaddress/uspsvalidation.js`

8/11/24: Latest js code for uspsvalidation:

`Frans/MultiShip/view/adminhtml/web/js/quoteaddress/uspsvalidation.js`

`design/frontend/Frans/default/Magento_Multishipping/web/js/billing-address-validation.js`

Magento dev tricks and tips

1/ To debug email sending code:

For example, when placing an order in FrontEnd; the order immediately gets marked as processed; and the quote is cleared. In order to debug sending email:

in file `vendor/magento/module-checkout/Controller/Onepage/Success.php`, disable this code:
`\$session->clearQuote();`

2/ Store object in memory per session

There is a deprecated technique still being used in Magento: use registry to store objects in memory.

Example:

```
$this->registry->unregister('current_creditmemo');  
$this->registry->register('current_creditmemo', $creditmemo);
```

PHP files and XML files to edit

- Shipment grid:

code/Frans/CustomOrderGridExport/Rewrite/Framework/View/Element/UiComponent/DataProvider/FilterPool.php

code/Frans/OrderProcessing/Model/ResourceModel/Shipment/Grid/CollectionShipment.php

code/Frans/OrderProcessing/view/adminhtml/ui_component/sales_order_shipment_grid.xml

Custom columns, such as

code/Frans/OrderProcessing/Ui/Component/Listing/Column/Shipment/InvoiceNumber.php

Coding guidelines

- JavaScript: do NOT translate text!

Instead of

```
$.mage.__('Attention')
```

just use

```
`Attention`
```

Frans is an English-only website for now.

- Don't use `INNER JOIN`
Use `JOIN`. Keep it simple. `INNER JOIN` is the same as `JOIN`
Don't use `LEFT OUTER JOIN`
Just use `LEFT JOIN`. Keep it simple. `LEFT OUTER JOIN` is the same as `LEFT JOIN`
- Table design, ID column:
Every table must have int identity unsigned pk column, named `id`

```
<table name="frans_stripe_reauth_log" resource="default" engine="innodb" comment="Stripe
Reauthorization Logs">
  <column xsi:type="int" name="id" padding="10" unsigned="true" nullable="false" identity="true"
comment="ID"/>
```

Keep column name simple. Instead of `log_id`, `order_item_id`, just name the column `id``.
The table name already tells if it's a log or an `order_item`. No need to repeat that in the column name.

- Table design text column:

Try not to use `text`` column; because it affects performance. Use `varchar()` with a limit instead. Only use `text`` column if it's really necessary, such as storing raw shipping label data, raw image data..

- For SQL statement, when using `GROUP_CONCAT` it's highly recommended to use `|` as separator. Because by default, most database use comma as a separator. When raw data is used for csv output; column cannot contain comma.
For example, instead of
`SELECT GROUP_CONCAT(order_id) -- which yield 3,5,7 -> breaks the CSV download`
we should do
`SLECT GROUP_CONCAT(order_id SEPARATOR '|') -- which yields 3|5|7 -> does not break`
for the CSV download

- For SQL statements, do NOT put column on its own line like this

```
SELECT
  qa_id,
  MAX(invoice_item_id) AS example_invoice_item_id,
  MAX(invoice_id) AS invoice_id,
  MAX(si_num) AS si_num,
```

because it takes forever to scroll up and down.

Write columns on the same line like this:

```
SELECT
  qa_id, MAX(invoice_item_id) AS example_invoice_item_id, MAX(invoice_id) AS invoice_id,
  MAX(si_num) AS si_num,
```

- SQL CASE:

For simple IF/ELSE, do NOT use CASE like this

```
CASE WHEN soi.product_notes IS NULL OR soi.product_notes = " THEN NULL ELSE soi.product_notes END
```

Instead, is IF (if MSSQL, use IIF):

```
IF(soi.product_notes IS NULL OR soi.product_notes = ",NULL,soi.product_notes)
```

Only use CASE WHEN if there are more than one if/else branches.

- To check if a value exists, use if (\$value !== null)
Do not use if (\$value)
because when \$value is zero, if (\$value) does not pass.
Example code:

```
if($quote->getIsMultiShipping() && $quoteItem->getCustomPrice()) {
  $item->setPrice($quoteItem->getCustomPrice())-
  >setBaseOriginalPrice($quoteItem->getCustomPrice());
```

This code is wrong. When customPrice is zero (no-charge), this if statement does not pass.

Rewrite it to this:

```
if($quote->getIsMultiShipping() && ($quoteItem->getCustomPrice() !== null)) {
  $item->setPrice($quoteItem->getCustomPrice())-
  >setBaseOriginalPrice($quoteItem->getCustomPrice());
```

now, if customPrice is zero, this if statement passes

- Class overriding/extending

When overriding or extending a class; make sure to document at the top of the PHP/Javascript file. For example:

```
<?php
/**
 * Frans overrides Techgroup\UspsValidation\Controller\Index\Address
 */
```

- Javascript

Each function must have documentation explaining its purpose, and one or a few usage example(s)

```
/**
 * Frans Address Validation module for BE
 * 2024/09/02 Brian created
 * Usage: define('addrval', addrVal)
 * addrVal.validateAddress
 */
define([
  'jquery',
  'ko',
  'Magento_Ui/js/form/form',
  'Magento_Ui/js/modal/modal',
  'Frans_Framework/frans/framework/axios.min',
], function ($, ko, Component, modal, axios) {
  'use strict';
```

- **Branching rules:**

When you work on a task, such as S1234, create a branch with the same name, such as branch s1234

For example, task <https://plan.socalappsolutions.com/view.php?id=S1904p4> --> branch name = s1904p4

Notice that git does not work well if we name the branch with numbers only; so do NOT name it with numbers only, such as `1904`. Sometimes it works, sometimes git will throw an error because it thinks 1904 is a commit sha, not a branch name.

Make sure branches are lowercase, not uppercase. For example, branch name = s1904, not **S1904**

- PHP code to get table name: no need to call functions to get table name, just use string literal. Instead of

```
resourceConnection->getTableName('frans_order');
```

Just type 'frans_order'

Dev bulletin 11/15

Currently there's a difference between FE and BE MS orders:

- for FE MS orders, the sales-ship-item stores product id of configurable-product; and store the SKU of the simple-product.
- for BE MS orders, the sales-ship-item stores product id of simple-product; and store SKU of the simple-product

Example: right is front-end, left is back-end. Same product orders; but data stored differently in database

	1	2
entity_id	518	517
parent_id	306	305
row_total	<null>	<null>
price	12.9500	12.9500
weight	0.2000	0.2000
qty	2.0000	2.0000
product_id	9859	9858
order_item_id	768	766
additional_data	<null>	<null>
description	<null>	<null>
name	16pc Smoked Salt Thins	Smoked Salt Thins
sku	THINS-SMOKED-SALT-16PC	THINS-SMOKED-SALT-16PC
accpac_sku	<null>	<null>
product_notes	<null>	<null>

To address this difference; in reports; we currently need to UNION two queries. For example, ItemsOnOrderByItem:

app/code/Frans/AdminReport/Controller/Adminhtml/Action/ItemsOrdByItemCsv.php

Query:

```
## # First set, pull from configurable prod. Because FE MS orders has
sales_shipment_item storing prod-id of CONFIGURABLE PROD, we need to union
this set. In the
# future we might want to rewrite either FE or BE to make sure they are
consistent
SELECT
# ,ss.entity_id,o.updated_at, o.entity_id,ssi.product_id
max(ssi.parent_id) example_shipment_id,max(ssi.product_id)
example_product_id
,ssi.sku as prod_sku,max(ce_vc.value) as product_name # 9/27
,max(aovcolor.value) color_title
,sum(ssi.qty) quantity #9/27 pull qty from configurable prod

FROM sales_shipment_item ssi
JOIN catalog_product_entity pconf ON pconf.entity_id=ssi.product_id AND
pconf.type_id ='configurable'

UNION
## Second set, pull from simple prod
SELECT
# ,ss.entity_id,o.updated_at, o.entity_id,ssi.product_id
max(ssi.parent_id) example_shipment_id,max(ssi.product_id)
example_product_id
,ssi.sku as prod_sku,max(ce_vc.value) as product_name # 9/27
,max(aovcolor.value) color_title
,sum(COALESCE(ssi_conf.qty,ssi.qty)) quantity #9/27 pull qty from configurable
prod, if any

FROM sales_shipment_item ssi
JOIN catalog_product_entity psimp ON psimp.entity_id=ssi.product_id AND
psimp.type_id ='simple'
```

Because FE MS orders has sales_shipment_item storing prod-id of CONFIGURABLE PROD, we need to union. In the future we might want to rewrite either FE or BE to make sure they are consistent.

Coding Suggestions

Here are suggestions when coding. These are less strict than [coding-guidelines](#).

1/ SQL select performance:

- If possible, always specify the list of columns you want to select. For example:

```
$this->connection->select()  
->from($mainTable, ['id'])  
->where('ent_type = ?', 'shipment')  
->columns('id') // ONLY id is needed  
->order('id DESC')  
->limit(1)
```

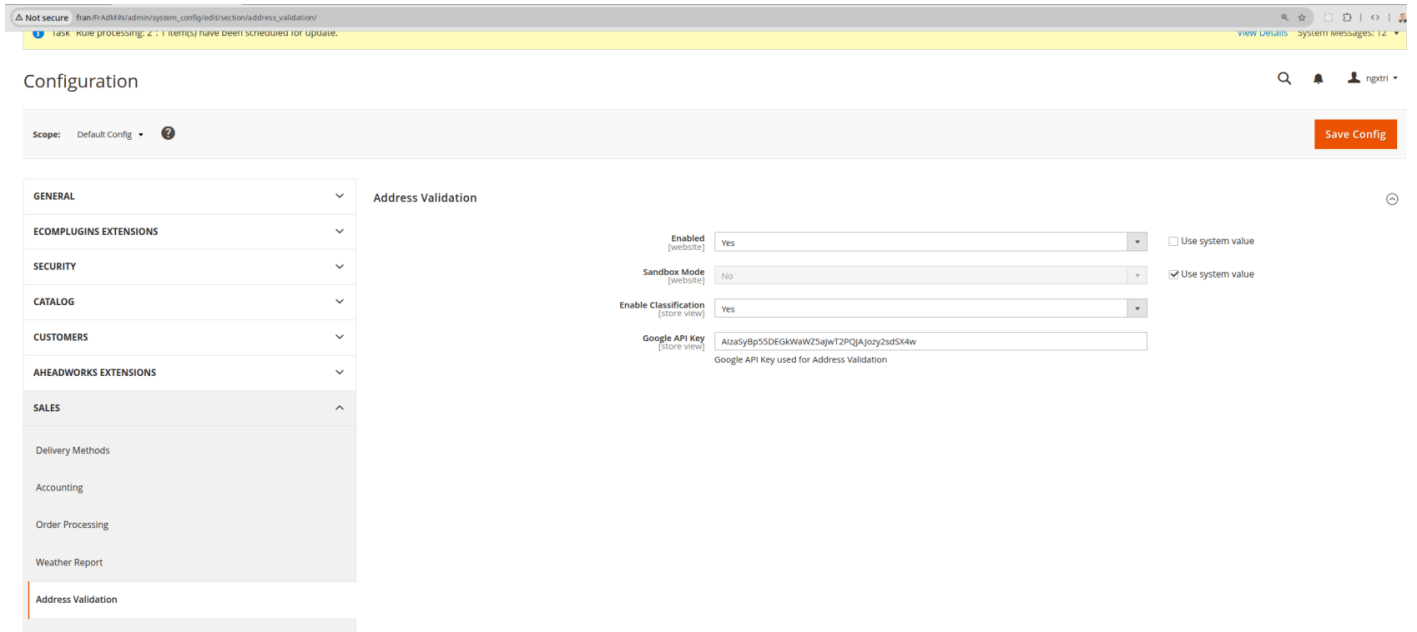
In the above example; since we only care about the count of rows; we only need to specify `columns('id')`

There is a big difference from querying 10 columns versus querying 2 columns.

Frans Extensions

- AddressValidation: app/code/Frans/AddressValidation

Config values:



The screenshot shows the Magento Admin configuration interface for the 'Address Validation' extension. The browser address bar indicates the URL: `fran.fr/ADMIN/admin/system_config/edit/section/address_validation/`. A yellow notification banner at the top states: 'Task: rule processing: 2 - 1 item(s) have been scheduled for update.' The page title is 'Configuration' and the scope is 'Default Config'. A 'Save Config' button is visible in the top right corner. On the left, a sidebar menu lists various configuration sections: GENERAL, ECOMPLUGINS EXTENSIONS, SECURITY, CATALOG, CUSTOMERS, AHEADWORKS EXTENSIONS, SALES, Delivery Methods, Accounting, Order Processing, Weather Report, and Address Validation (which is currently selected). The main content area is titled 'Address Validation' and contains the following configuration options:

- Enabled** [website]: Yes Use system value
- Sandbox Mode** [website]: No Use system value
- Enable Classification** [store view]: Yes
- Google API Key** [store view]: AlzaSyBp55DEGKwWZ5ajwT2PQAJozY2sd5X4w
Google API Key used for Address Validation