Mage TITANS

MCR 2016

# TADHG BOWE

Magento Lead Developer at Screen Pages

@tadhgbowe

# Migrating Data from Magento 1 to Magento 2

WWW.MAGETITANS.CO.UK

#MageTitansMCR @MageTitans

# Who am I?

- My name is **Tadhg**. I’m Irish! That’s all you need to know for now.

# Where I work. Some Screen Pages key facts

- 300+ e-commerce websites since 1997
- Magento specialists since 2009 (ca. 100 implementations)
- Magento 2 certified
- It’s a cool place to work ☺

screen pages.

MAGENTO GOLD

MAGENTO GOLD SOLUTION PARTNER 6 YEARS OF MAGENTO

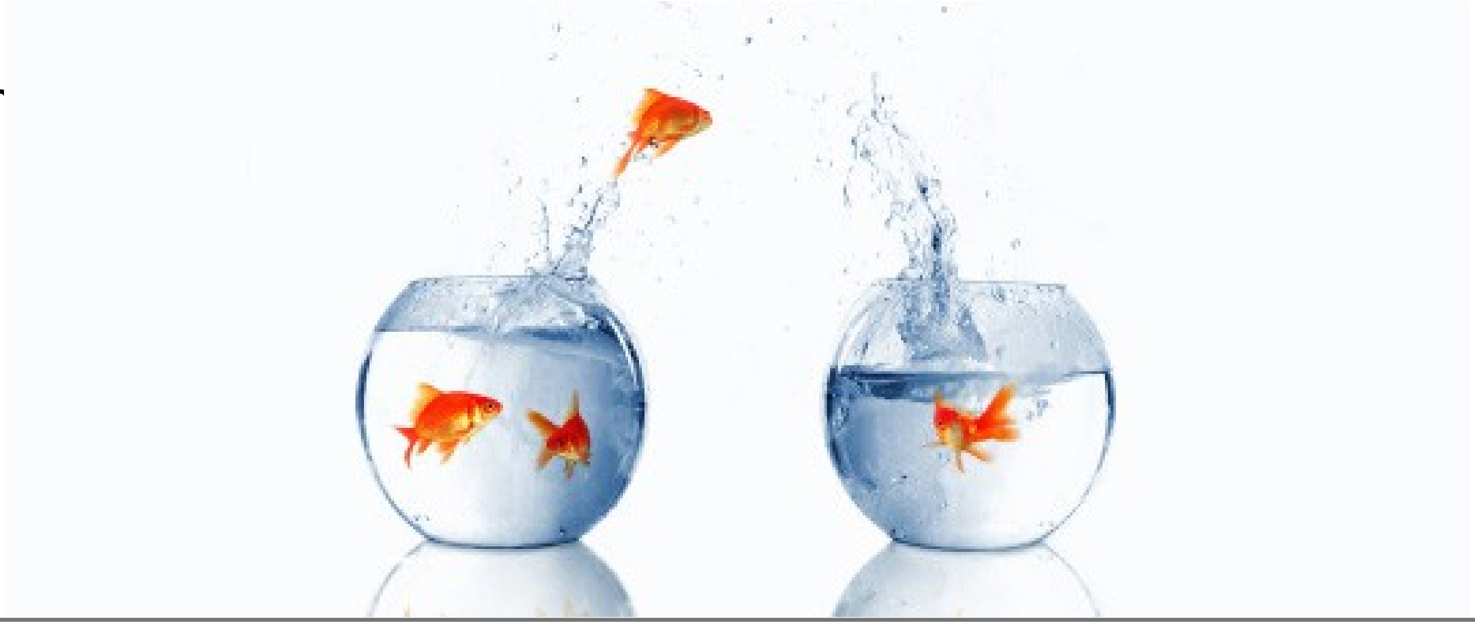
venda ecmod direct commerce awards Winner 2013

# THANK YOU FOR YOUR FEEDBACK…

# So what's the story then? The agenda is:

- Why talk about Data Migration!
- What is it and how does it work.
- How do we plan for it.
- What actually happened when put to the test!

- Ar



# Why stand up here and talk about Data Migration?

- Yes we actually have an Enterprise client migrating from Magento 1 to Magento 2. Here are some stats:

- Magento 1 version 1.13.0.2

- 21000+ customers
- 5000+ products
- 45000+ sales orders
- 100000+ quotes

- 15+ third part extensions
- 40+ modules written in-house

# Early on…

- When the M2 project kicked off "data migration" was the hot topic of conversation!

- We knew Magento had a tool already available to help. So we had to find out did this data migration tool from Magento actually work?!

- Confession: I have not looked into any other migration tools (well okay I did try one but it crashed). But if Magento produces one you'd be pretty confident it works right?! ☺

- **QUESTION:**

  How many people have used/tested the data migration tool?

# How Magento’s Data Migration Tool works:

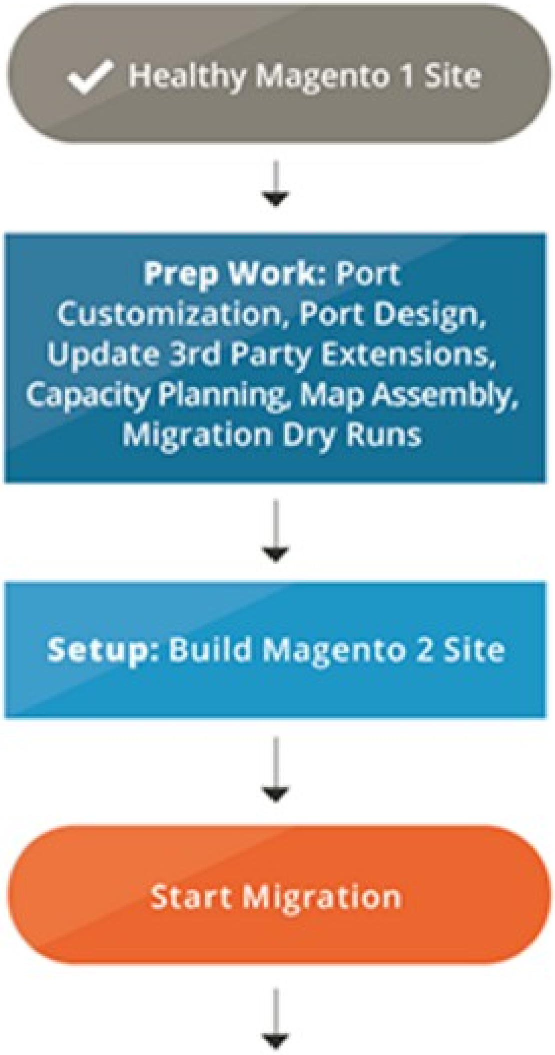
- RTFM – Read The Fantastic Manuals (and take your time!):

  http://devdocs.magento.com/guides/v2.0/migration/bk-migration-guide.html

- The migration tool is split into three phases (or “modes”):

1. Configuration settings
2. Data
3. Delta / Incremental (that is, changes since the last data run)

# Steps to migratng data:



✓ Healthy Magento 1 Site

**Prep Work:** Port Customization, Port Design, Update 3rd Party Extensions, Capacity Planning, Map Assembly, Migration Dry Runs

**Setup:** Build Magento 2 Site

Start Migration

**Important:**
3rd Party Extensions
Migration Dry Runs

# To install:

- To Install Magento’s migration tool:

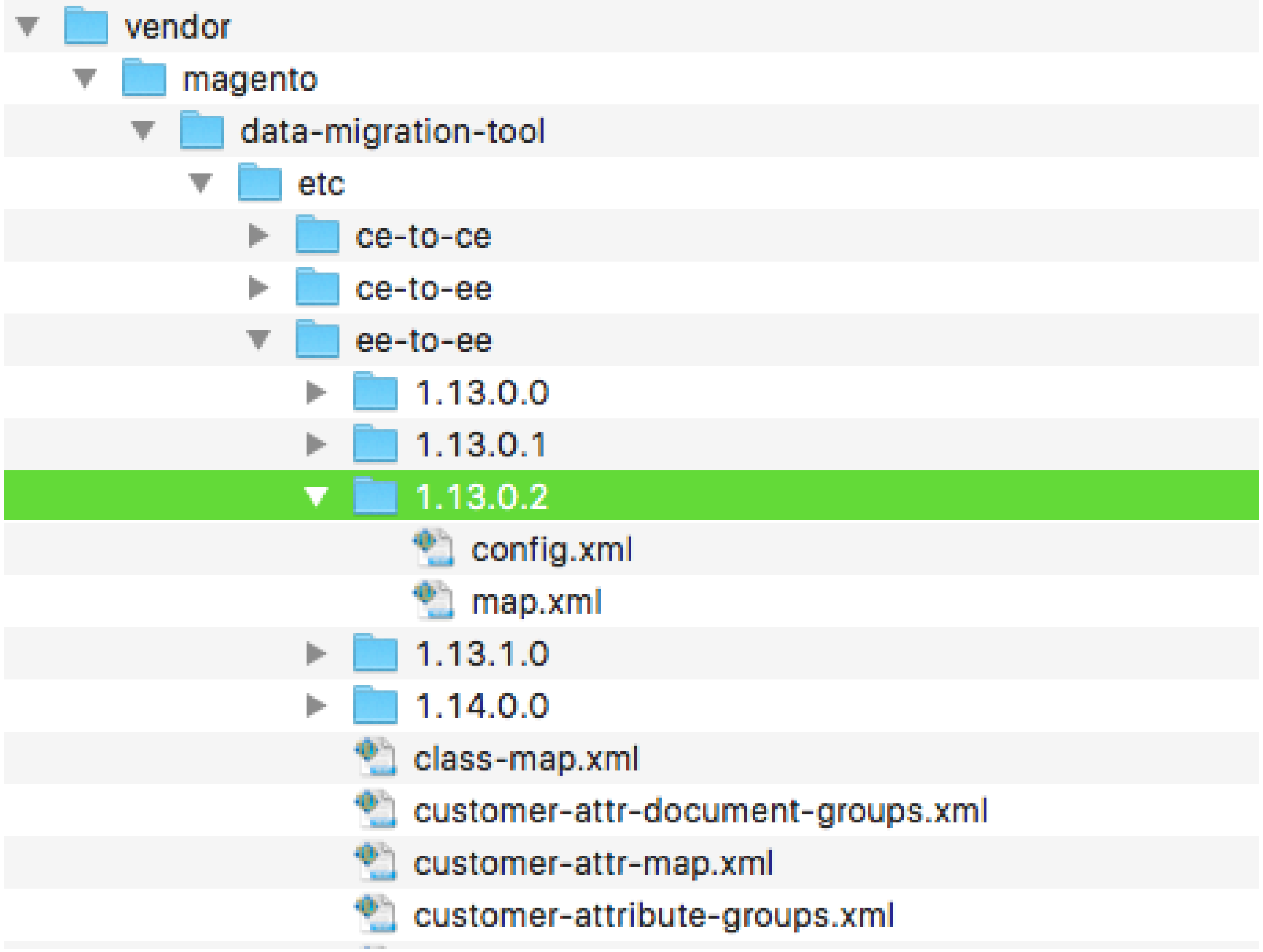
  ```
  > composer config repositories.data-migration-tool git
  https://github.com/magento/data-migration-tool
  ```

  ```
  > composer require magento/data-migration-tool:2.1.0
  ```

- Please Note: Straight after v2.1.0 was released the data migration tool was not available. It took about 7-10 days for it to become available. So you might see this:

  “The requested package magento/data-migration-tool could not be found in any version”

# How to configure/run the Data Migration Tool:

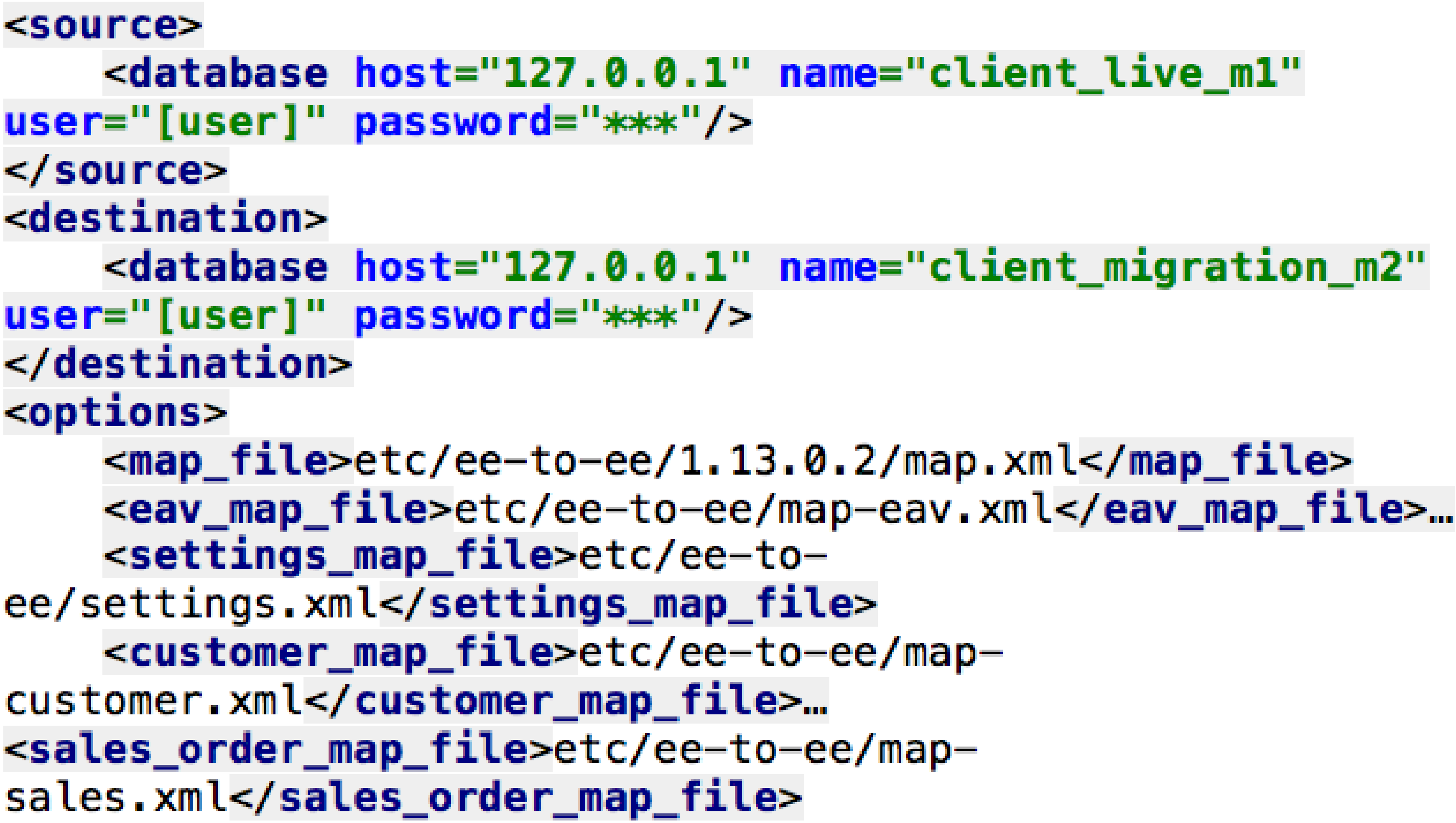


```
▼ vendor
  ▼ magento
    ▼ data-migration-tool
      ▼ etc
        ▶ ce-to-ce
        ▶ ce-to-ee
        ▼ ee-to-ee
          ▶ 1.13.0.0
          ▶ 1.13.0.1
          ▼ 1.13.0.2
              config.xml
              map.xml
          ▶ 1.13.1.0
          ▶ 1.14.0.0
            class-map.xml
            customer-attr-document-groups.xml
            customer-attr-map.xml
            customer-attribute-groups.xml
```

# How to confgure/run the Data Migration Tool:

- Find the version folder you are migrating from.
- Create a config.xml file (rename the .dist file already there).
- In config.xml:
  - Make note of the steps defined (settings, data, delta).
  - Define the source and destination databases
  - Decide which options apply. That involves renaming the mapping files you need.

# Example config.xml file:



```
<source>
    <database host="127.0.0.1" name="client_live_m1"
user="[user]" password="***"/>
</source>
<destination>
    <database host="127.0.0.1" name="client_migration_m2"
user="[user]" password="***"/>
</destination>
<options>
    <map_file>etc/ee-to-ee/1.13.0.2/map.xml</map_file>
    <eav_map_file>etc/ee-to-ee/map-eav.xml</eav_map_file>…
    <settings_map_file>etc/ee-to-
ee/settings.xml</settings_map_file>
    <customer_map_file>etc/ee-to-ee/map-
customer.xml</customer_map_file>…
<sales_order_map_file>etc/ee-to-ee/map-
sales.xml</sales_order_map_file>
```

# How to run the Data Migration Tool:

- **Settings** - stores; websites; and different system configuration (core_config_data):

```
> bin/magento migrate:settings /sites/m2migration/vendor/magento/data-migration-tool/etc/ce-to-ce/1.9.2.4/config.xml
```

- **Data** – eav, categories, products, customers, orders, quotes, wishlists, ratings, reviews etc...

```
> bin/magento migrate:data /sites/m2migration/vendor/magento/data- migration-tool/etc/ce-to-ce/1.9.2.4/config.xml
```

- **Delta** - migrate only the changes made in Magento 1 since the last time you migrated data.

# Planning Data Migration:

- First of all we had to find out does it work!?

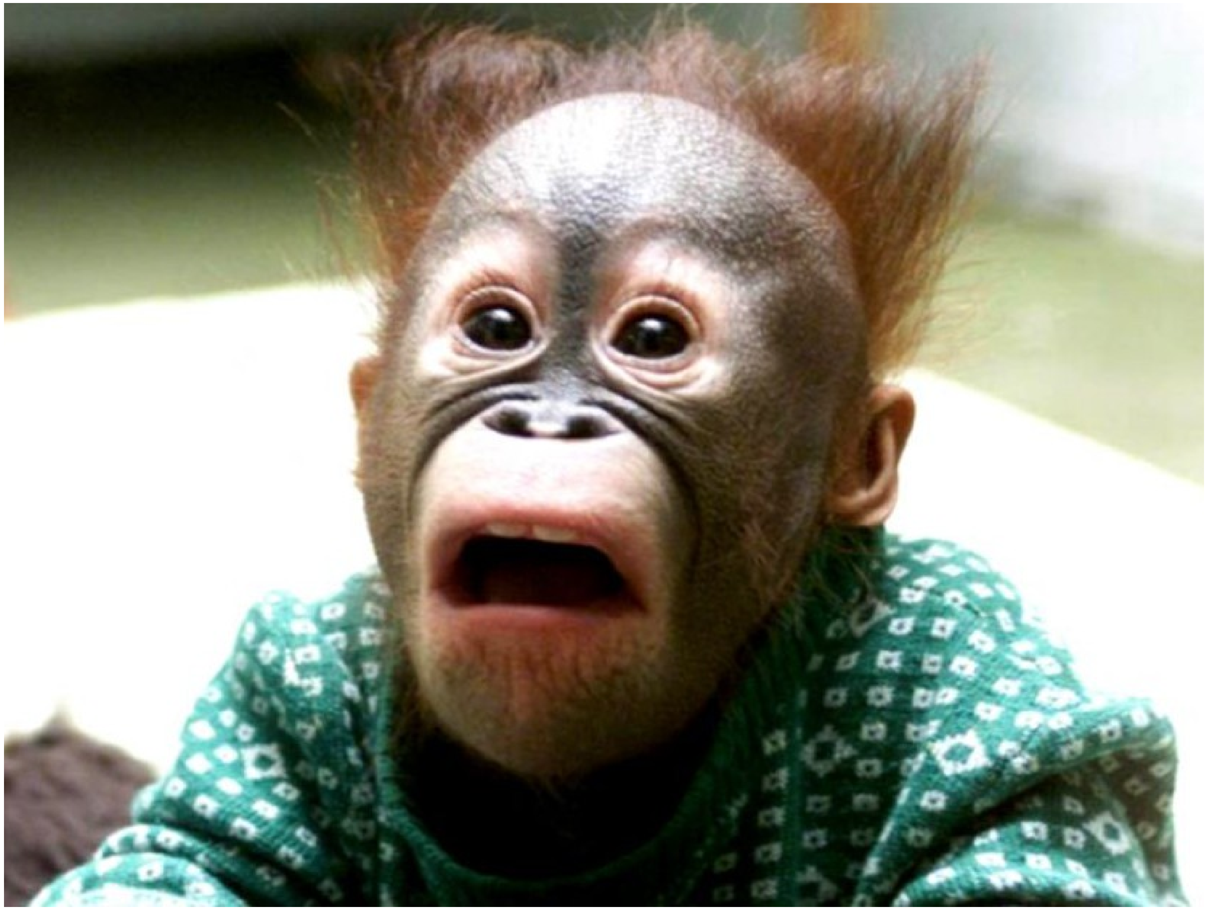
  Created a new M2 2.0.7 CE and M1 1.9.2.4 CE.
  In M1 we created some extra store views, configurable products, customers and generated some sales orders.
  **And yes! It all worked. We can use it ☺**

- So when do we run the data migration? What's the best plan of attack? Start testing early!

- Does M2 destination database need to be empty? All the entity IDs like Customer Entity ID, Product Entity ID need to carry across exactly the same.

- Take backups of your database (source and destination). Take backups as you go along too e.g. after settings migrate, after data

# Happy so far? :-0



# Now the fun begins!!

- So our real world client – M1 1.13.0.2 to M2 2.1.0. Let’s go!

- Settings: Yes migrated fine!

- Data: No! Okay so what errors did we get:

  Source documents are not mapped: aw_core_logger,aw_followup_config…

  Destination documents are not mapped: magestore_storelocator_holiday,magestore_storelocator_image…

  Source fields are not mapped. Document: catalogrule. Fields: points_action, points_currency_id…

# There's more…

Destination fields are not mapped. Document: admin_user. Fields: tfa_login_enabled, tfa_login_secret…

- These can all be fixed in map.xml. You can decide to "ignore" fields completely or "move" them. Others like the following:

Source fields are not mapped. Document: sales_flat_order. Fields: exported,export_attempts…

Destination fields are not mapped. Document: sales_order. Fields: prima_export_status

Those can be fixed in the map_sales.xml. Handy tool here: http://oyenetwork.com/articles/data-migration-to-magento2-case-study/

- Once you've mapped all those tables and fields in the correct xml files, rerun the data migration command and off we go again!

# Continuing on…

- Here’s another issue:

  SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry ’first.last@somewhere.com' for key 'CUSTOMER_ENTITY_EMAIL_WEBSITE_ID'

- The above meant we had duplicate customer email addresses for the same website in the source DB. Oh Oh!

- This was fixed by simply deleting the unwanted accounts via the M1 source Admin Panel.

- Okay it’s not nice seeing these but at least the migration tool picked them up. Clean data remember!

# But then this...☹

- But then:

  [Exception]

  Notice: Undefined offset: 1 in /
.../sites/client_site/vendor/magento/data-migration-tool/src/Migration/Step/Eav/Data.php on line 433

- The above is related to function moveAttributeToGroup??? After some investigation:

  **That horrible sinking feeling of**

  **OMG IT’S NOT GOING TO WORK...**

# Massive rethink!

- So the source EAV attribute table contains over 800 rows. That’s not normal for a Magento 1.13 install. Oh oh it’s an upgrade! And it looked like an upgrade from a CE v1.6 (ish).
  Is that compatible… well...

- Should we try continue and shoehorn all this data in? It’s not really how M2 structures its EAV nowadays. Nothing matches up.

- **So let’s be “Data Smart”**. We want clean data, data that M2 expects to see. Let’s avoid junk! M2 doesn’t need any more headaches to deal with ☺

- So we reviewed all the source data. What were the must haves? What data is the client reviewing themselves? Is there data we don’t actually need to migrate across? So rather than “hitting and hoping” on the data migration step we’re now approaching it correctly. Maybe it was a blessing in disguise that it failed!

# Let’s be data smart…

# Yes let’s be Data Smart…

- So what did we do? Here were the main players:
- **EAV not being migrated. It simply didn’t work.**
- **Catalog Products and Categories:** Being reviewed by client. This project was a redesign too. For example things like product images were being revisited. Okay lets reimport via Product Import. Not using data migration.
- **Reviews:** Import via a Magento 2 third party extension. M1 export extension installed.

- **Quotes:** Ignored.
- **CMS pages and blocks:** Ignored.
- **Wishlist:** Ignored.

---

- **Only migrating Customers, Newsletters, Sales Orders (including shipments, invoices & credit memo) Cool!**

# How was this achieved:

- Edit config.xml file and commented out the steps we no longer wanted e.g. EAV step, Ratings step, Tier Prices step.

- Migrate data into separate empty M2 database. Then copy only the files we need across to go-live database.

# Success! But a few little tweaks…

- **So it worked for us!** We only copied the DB tables we needed i.e. Customers, Sales Orders, Newsletter.

- But hold on! What about Customer EAV for example? What about any bespoke customer attributes in an M2 third party module that also existed in M1!?

- For example we have a customer third party EAV attribute “integration_customer_number” (type varchar). It exists in our source and destination. The data migrates across. You just need to make sure the attributes IDs match up. Why? The migration tool copies across the data assuming the EAV attributes have also come across. In this example our final go-live DB attribute_id was 177. The source attribute_id was 987!

# Some more notes…

- By using the main GIT master (go live) branch on the migration DB it ensures all DB fields are in place before data migrated. And if everything is mapped correctly (xml files) then it should go pretty smoothly.
- In the end we decided not to run the Migration Settings step! One of our store views in M1 was being moved to a new website in M2. Another store view was being removed completely. So why bring across settings we no longer need. It also gave us an opportunity to take a fresh look at all the client's settings – and it's a chance to learn the new M2 Admin Panel ☺

- How long did the data migration step take: Roughly 16 minutes (Mac Mini 3.0 GHz Intel "Core i7" 8GB RAM PHP 7.0.9 Percona MySql 5.6.25).
- That means for go live deployment: no Delta migration step. The data migration step can easily be completed while a holding page is up ☺

# Anything else?

- Sales Order Increment ID. There’s a new sequence table for sales orders split by Store ID e.g. sequence_order_1. Edit this table and set the increment ID to a value past the last Live order taken. Same applies for invoices, shipments etc…

[EDIT – this should work if you migrate settings + data – always check]

- Just in case: If your M1 site used a prefix then in M2 that’s set in sales_sequence_profile table which is linked to the sales_sequence_meta table.

- I mentioned we had a shake up of store views. One of our M1 store views no longer exists (it was a mobile view). Just be careful copying across data where Store IDs may no longer match up.

- So what about our products and categories?

# Getting ready to party…???

# Nearly there!

## How do we migrate products and categories?

- Export all products from M1 using built in System > Export. In our case they were all configurable & child simple products.

- Product Importing in M2 has changed. It’s now one row per product. For configurable products we have a very busy column called configurable_variations. For example you have something like:

```
sku=ASHCH060,display=1,size=6|
sku=ASHCH065,display=1,size=6.5|…
```

# Will this talk ever end…

- Configurable product dropdown attribute values need to be imported first! Look at sample data modules e.g. module-catalog-sample-data, module-configurable-sample-data for example CSV files!

- Product Reviews: We installed a free extension on our M1 site so we could export reviews in the exact format that a purchased M2 extension needed for importing.

# In Summary:

Think Data Smart

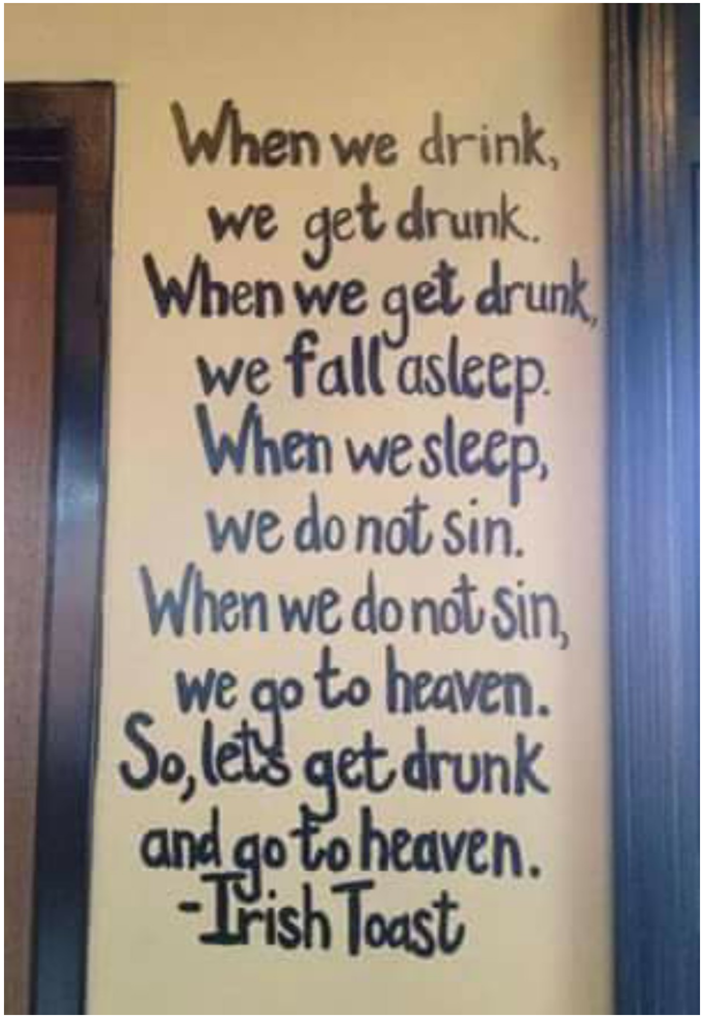
Allow plenty of time!

Plan well ahead

There are no two Magento 1 sites the same!

The Magento team are human too.
It’s very easy to be negative.
But there’s a whole lot to be positive about.

# Finally…I shall leave you with an Irish Toast:



When we drink,
we get drunk.
When we get drunk,
we fall asleep.
When we sleep,
we do not sin.
When we do not sin,
we go to heaven.
So, let's get drunk
and go to heaven.
-Irish Toast

screen pages.

MAGENTO GOLD