



# Ingram Micro Csv importer for Magento

## Project goal

Import csv data from Ingram Micro ftp into Magento catalog using magmi importer by cronjob

## Ftp download

The module will download the csv catalog from Ingram ftp. Ftp access will be configured into Ingram magento module (script with configuration parameters is provided).

The cronjob will be configured into magento configuration module.

The module will connect and download the csv file into magento “var/export” directory

All the php scripts will be hosted into “ingram” directory, into the magento root directory

## Main Csv Conversion

**ATTENTION: Every field in the ingram csv should be trimmed! Since it contains extra spaces.**

Ingram csv is a comma separated csv file without header, but the csv columns will follow the order shown below. Ingram will put in ftp 2 kind of csv: the first is a Total csv (with about 60k rows), the second is a delta csv which means it will contain only the difference between the actual product stock and the day before.

From this csv we need to create a new csv called “magmi\_import\_ingram.csv” into “var/import” dir, with header and value as the following mapping:

ACTION INDICATOR: can be “A”, “D”, “C”; means that the record is Addedd, or Deleted, or Changed.

INGRAM PART NUMBER → sku

VENDOR NUMBER →N/A

VENDOR NAME → manufacturer

INGRAM PART DESCRIPTION LINE 1 → short\_description

INGRAM PART DESCRIPTION LINE 2 → + short\_description

RETAIL PRICE → N/A

VENDOR PART NUMBER → mpn

WEIGHT → weight

UPC CODE → ean

LENGTH → N/A

WIDTH → N/A

HEIGHT → N/A

PRICE CHANGE FLAG → N/A

CUSTOMER PRICE → cost

STATUS → N/A

AVAILABILITY FLAG → is\_in\_stock, if Y = 1, if N = 0

CLASS→ N/A

ALLIANCE FLAG→ N/A

CPU CODE→ N/A

MEDIA TYPE→ N/A

categories → from configuration table, look up the value INGRAM MICRO CATEGORY / SUBCATEGORY code and put the magento category, see Categories section for details

NEW ITEM RECEIPT FLAG → N/A

COUNTRY CODE → N/A

In addition to these fields the csv will contain other fields as follow

distributor → “ingram”

markup → from product attribute “markup”, if attribute is empty or product doesn't exist in magento database, will use category configuration markup, if category configuration doesn't exist, will use configuration default markup and send an email to the address configured in Notification Email Address (magento module configuration).

markup → from product attribute “markup”, if attribute is empty will use category configuration markup, if category configuration doesn't exist, will use configuration default markup and send an email to the address configured in Notification Email Address

price →  $\text{cost} * (1 + \text{markup}/100)$

priceamzit →  $\text{price} * (1 + \text{MarkupAmzIt}/100)$

priceamzfr →  $\text{price} * (1 + \text{MarkupAmzFr}/100)$

priceamzes →  $\text{price} * (1 + \text{MarkupAmzEs}/100)$

priceamzuk →  $\text{price} * (1 + \text{MarkupAmzUk}/100)$

priceamzde →  $\text{price} * (1 + \text{MarkupAmzDe}/100)$

## Magento module configuration parameters

Ftp Host

Ftp User

Ftp Password

Ingram Price Filename ( put the complete path i.e. /FUSION/IT/TWBD1R/PRICE.ZIP )

Ingram Stock Filename (put the complete path i.e./FUSION/IT/AVAIL/TOTITHRL.ZIP)

Ingram Categories (put the complete path i.e. /FUSION/IT/NEWCATS/NEWCATZ.ZIP)

Categories Mapping csv ( see categories mapping section for details)

Default Markup (%)

Default Category Status default OFF

Import Out of stock products (yes/no) default NO

Notification email address

Time out to delete products

Closed for Holiday YES/NO

## Categories mapping

Ingram categories will be imported into a specific table called “ingram\_original\_categories”. This table will be made of 3 fields: item\_desc, item\_cat\_code, item\_subcatcode. If the category table is not present in magento database, the script will create it.

Everytime the NEWCATZ.ZIP will be downloaded (every day), this table will be truncated and the new file will be imported. For more stability, the old table will be renamed to “ingram\_categories\_old”. The file will be imported into a table with the same structure. If the number of records is more than 500 the new table will remain. If not, the new “ingram\_categories” table will be dropped and the old table will be restored.

The categories cronjob will be executed before PRICE.TXT cronjob, because this way we will be sure every product will find his own category.

When the PRICE.TXT cronjob will be executed, for every row we will look up to this table. PRICE.TXT contains the Ingram category and subcategory codes, so we will look for them to find the category name (item\_desc). To get the complete category/subcategory name we have to look 2 times into the table. The first time we will search for the category code, the second time we will search for category and subcategory code.

This is a sample of category file NEWCATZ.ZIP

"COMPUTER SYSTEMS ", "00", "\* "

"DESKTOPS/TOWERS ", "00", "01"

"COMPUTER&MNTR UNITS ", "00", "02"

"SERVERS ", "00", "05"

"ULTRAMOBILE PC(UMPC)", "00", "10"

so if we have the following code “0002” in PRICE.TXT, into the category/subcategory field, than the category/subcategory name will be “COMPUTER SYSTEMS/COMPUTER&MNTR UNITS”

## Category mapping table

The categories mapping “ingram\_categories” table will be created by our magento module and will have the following columns:

CategorySubcategory

Magento Category (i.e. “telefonía – tablet/telefonía – navigazione/smartphone – pda phone”)

Markup

MarkupAmzIt

MarkupAmzFr

MarkupAmzEs

MarkupAmzUk

MarkupAmzDe

Status (0 or 1)

The first import will populate the entire table with the default values. The admin will then download and upload into the magento system configuration a csv file to update this table.

When converting the original PRICE.TXT csv in magmi csv, we will loop on every row checking first of all the Status. If for that product category the status is zero, the product will not be imported.

Then we will check the Magento Category to put into categories field. The prices will be calculated as stated above, applying the markups for each price accordingly.

## Notification Email

If the Ingram category is not found into the ingram\_categories table, a notification email will be sent.

This email will include all the product rows not found in the body and will have as subject "New Ingram Categories found"

## Module in Action

The module will have to download and decompress the PRICE.ZIP file to get the final PRICE.TXT file. The module will basically convert a csv to another csv with some calculation on prices and category, applying configuration data where needed. The module will have to check all the products coming from Ingram distributor because of the markup attribute, but will check the products not updated too. To accomplish this task we will provide a magento collection that will extract all the ingram products not updated since last 24 hours that have non zero quantity and will put them to zero qty (magento collection php code is provided). This routine will create a new file in var/import called "magmi\_zero\_ingram.csv": this file will have 2 columns "sku,qty" where sku is the product sku (INGRAM PART NUMBER) and qty is always zero.

To clean up the database from old products we will check all the products with zero quantity not updated since the last x days configured in "Time out to delete products" field and will put them into the magmi\_delete\_ingram.csv

This csv will contain only 2 columns:

sku → sku

magmi:delete → 1

If there aren't products to delete, the file will left untouched.

## Closed For Holiday

If this field is 1, the PRICE.TXT import will be skipped and all the skus on magento will be set to zero qty in magmi\_zero\_ingram.csv. The timeout flag will be ignored.

## Test Cases

1. Check if category not found email is sent: create a new row with dummy category in PRICE.TXT and check if mail is sent with the right content
2. Check if products not in csv are setted to zero qty: create a dummy product in PRICE.TXT with 100 qty, than remove it and check if after 24 hours it will be at zero qty
3. check if magmi\_delete\_dialer.csv is created for products not updated: take an old product in magento catalog, set the “distributor” attribute to “dialer”, than check if the magmi\_delete\_dialer.csv is created.
4. Test the holiday flag to 1 behaviour