|  |
| --- |
|  |
| Product Data Extracts  DESIGN |

|  |  |
| --- | --- |
| Company | IPEX Management Inc. |
| Project Title | PIM (Product Information Management) |
| Project Number |  |
| Date | March 2, 2016 |
| Version | 1.2 |



**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| Version | Modifications | Date / Name |
| 1.0 | Initial version | Feb.22, 2016 – Vianney Rebetez |
| 1.1 | Review of customer report and trade services | Feb.29, 2016 – Vianney Rebetez |
| 1.2 | Review of customer report format | Mar. 2, 2016 – Vianney Rebetez |

**TABLE OF CONTENTS**

[1 DATA EXTRACTION 4](#_Toc444703227)

[1.1 Data extraction options 4](#_Toc444703228)

[1.2 Oracle DB views 4](#_Toc444703229)

[1.3 Product data extracts 6](#_Toc444703230)

[2 IPEX CUSTOMER STANDARD PRODUCTS REPORT 6](#_Toc444703231)

[2.1 Format 6](#_Toc444703232)

[2.2 Content 6](#_Toc444703233)

[2.3 Process 7](#_Toc444703234)

[3 TRADE SERVICES 8](#_Toc444703235)

[4 INDUSTRY DATA WAREHOUSE – IDW 8](#_Toc444703236)

[5 USER PRODUCT DATA EXTRACTOR 8](#_Toc444703237)

# DATA EXTRACTION

The goal is to extract product data for customers and data pools. These data extractions are not a continuous outbound integration process but more of a reporting that needs to happen at regular interval or on demand.

## Data extraction options

For these reporting purposes, Stibo and IPEX considered 3 possible solutions: outbound integration to SQL Server DB, Mongo DB and Oracle STEP SQL API. Stibo strongly recommended the Mongo DB option as it is freeware and they have developed a fully-integrated connector with STEP. They also strongly advised against the Oracle SQL API solution as they seemed afraid it direct querying on STEP DB could impact performance of STEP and that it may complicate DB maintenance and administration.

IPEX’s team didn’t feel as strongly against the Oracle DB option as it was using hardware and software already in place and would allow building the data extracts using plain and simple SQL. On the opposite Mongo DB was requiring extra licensing for the connector with STEP, extra virtual environment setups to install Mongo DB and an unconventional architecture for IPEX (JSON, java-scripting ...). Our decision is to go with Oracle SQL API during this first Phase of the project, keep an eye on system’s performance, see how product data extract requirements evolve in the future, and maybe reconsider the Mongo DB option at a later stage.

## Oracle DB views

STEP Oracle SQL API provides a set of views and functions to access the STEP DB in a very generic way. It provides views on nodes (products, classifications, entities, assets), links (references between nodes), attributes (either for links or nodes) and values. To make things easier for our data extraction requirements, present and future, we will first build an extra layer of views which will present our data in accordance to our data model.

* Products: all the product attributes and simple links (etim and unspsc classifications + infoflo product line) ... one row per product (key = product code)
* Product-Companies: all product-company attributes (pricing + last sold date) and simple links (brand / sub-brand + marketing product line) ... one row per product-company (key = product code + company code)
* Product-Sources: product code + source code + type of source (plant or supplier) + type of link (primary or alternate) + link to source (plant or supplier classification) ... one row per product-source (key = product code + source code)
* Product-Source Certifications: list of all certifications per product-source (certified standards and standards met) ... one row per product-source-certification (key = product code + source code + standard + agency)
* Packaging: list of all packages available with links to a package code and to the children of the package (either a product code or the I2of5 of another package) ... one row per package (key = I2of5)
* Product Packaging: packaging branches per product, starting with the product and showing successive packaging options (e.g. unit->carton->pallet->truck) ... one row per product-branch (key = product code + level 1 I2of5, only one row per product with default brand = YES)
* Package Codes: all possible package codes used by packaging with default dimensions (key = package code)
* Product Assets: all assets applicable per product, including assets of the material hierarchy node the product may be linked to (msds, crgs, line drawings, images) ... one row per product-asset (key = product code + asset id)
* Product-Company Assets: all assets applicable per product-company, including all assets of the marketing hierarchy branch(es) the product-company belongs to (if any) (brochures, install sheets, data sheets, case studies, info bulletins, technical manuals, letters of compliance, catalogues, terms & conditions) ... one row per product-company-asset (key = product code + company code + asset id)
* Product-Source Assets: all assets applicable per product-plant or product-supplier (certification files, 2d & 3d drawings) ... one row per product-source-asset (key = product code + source code + asset id)
* Marketing Hierarchy: a flat view of this hierarchy, by product line, giving for each the application, the market segment and the company (key = product line id)
* Infoflo Item Hierarchy: a flat view of this hierarchy, by product line, giving for each the product group, the super group, the product type and the market segment (key = product line id)

To minimize potential Oracle DB performance and administration issues (see section 1.1) and because some of the views above may be very long to run, we will adopt a product data warehouse approach. Once a day, data will be extracted from STEP Oracle DB to an SQL Server DB, so all the product data extracts discussed below will be based on data that can be one-day old.

## Product data extracts

In Phase 1 of the project, we will only extract product data to a generic format and not look into specific requirements of individual customers or data pools except for one: IDW (IDEA Industry Data Warehouse).

As it is currently the case, a generic Excel format will be used to provide customers and Trade Services, data about our products. The generic format however will be similar but not identical to the one currently in use.

Other data pools, Commport and Data Agility, have also been discussed but they won’t be covered specifically.

The next sections describe the requirements and design for 4 product data extracts

* IPEX Customer Standard Products Report
* Trade Services
* Industry Data Warehouse (IDW)
* User Product Data Extractor (Excel tool)

# IPEX CUSTOMER STANDARD PRODUCTS REPORT

## Format

The customer standard products report is an Excel workbook divided in 3 spreadsheets:

* Product Data: most product and product-company attributes and classifications plus product-source certified standards (exact list of attributes to extract will be defined separately) ... one row per product
* Packaging Data: for each product included in product data, packaging options (package type, package code, i2of5 and dimensions) presented as a succession on one row (e.g. product -> inner carton -> outer carton -> pallet -> truck) ... one row per product packaging branch
* Product Assets: for each product included in product data, all assets applicable ... one row per asset

## Content

* One report contains products linked to one company: 01 (ipex inc), 02 (ipex usa), 03 (multifittings corp) or 16 (ipex electrical).
* No restriction is made on market segments to be included or excluded.
* For companies 01 and 16, both French and English information is required and will be included in the same report while reports for companies 02 and 03 will be only English data.
  + A language column will be added to the Product Assets sheet to differentiate French and English assets.
  + Imperial and metric product description fields as well as marketing product line description and features & benefits will be differentiated per language but not the product characteristics (aka attributes of Infoflo product description generator, e.g. material, class, color, type ...)
* The following products should be excluded
  + Obsolete products (Status=O)
  + OEM products and regular products linked to OEM brands
  + Products not sold in past 3 years AND either [no current list price OR current list price effective for more than 3 years]
* Pricing data should not be included
* If a product has a product-level MSDS, we should not extract the material-level MSDS. In other words, a product should have only one type of MSDS: product-MSDS in priority, material-MSDS otherwise.
* Only one 2d drawing is required per product, whatever the number of sources and number of drawings available per source. No specific requirement about which one to select.

## Process

This report will be generated automatically on the 1st day of each month or anytime a change is registered on one of the following

* Certification (certified standards only)
* Current list price
* Marketing assets (excl. images)?

If one of the above changes happens less than X days previous to 1st of month, then the next “regular” update should be skipped.

When a new report package is created, either because of date or because of key change (see above), it should be saved to a determined location and an email should be sent to the business owner of the report for him/her to review the package before making it available to its recipients. The location where to store the package is most likely an FTP site accessible to customers who will be downloading it.

The product assets referenced in the assets section of the report will be extracted from their storage location (the asset library) and packaged together with the report in one single zip file. Many products can have references to the same asset but one asset will be extracted only once. When extracted from asset library, asset files need to be renamed using the asset Long Name (attribute of the asset) or Name if Long Name is not defined. The asset library path should be removed from the report after assets have been extracted and renamed. The package should be encrypted with a password.

# TRADE SERVICES

The report for Trade Services has the same requirements and design as the IPEX Standard Customer Report described in the previous section. It will use the procedure and result in the same report format.

Only the following exceptions/differences apply

TBD: Are all company + language combinations required?

TBD: For each company + language combination required what market segments should be included or do we need to break down reports per market segment?

TBD: Do we exclude all obsolete products or only those that have obsoleted in past X months?

Current list price and current list price effective date need to be included

# INDUSTRY DATA WAREHOUSE – IDW

To be completed ...

# USER PRODUCT DATA EXTRACTOR

We will implement an easily configurable product data extraction tool in Excel for IPEX users to extract their own product data. The tool will first let the user make a selection of products to extract using the following criteria

* One language (English or French)
* One company (01, 02, 03, 16)
* One or multiple market segments (By joining products and product-companies views using one company we get 1 Infoflo product line and 0 or 1 marketing product line. If any line maps to a market segment selected, the product is included.)
* Exclude obsolete products since 0, 3, 6, 9,12 months (selecting 0 months would mean to exclude all obsolete products)
* Exclude/Include OEM products (Exclude all products under OEM brands and all OEM products under regular brands)
* Exclude/Include products with no current list price
* Exclude products not sold since 1, 2, 3, 4, 5 years and not created since 0,3,6,9,12 months

The result should be presented in 3 sections similarly to what we defined previously for standard customer reports (see section 2) but the user should be offered to include or exclude the packaging and the asset sections. The user should also have the choice of attributes to include in the product data section.

The excel spreadsheet will also include a macro function to retrieve all asset files from where they are stored (asset library drive), rename them to either Name or Long Name (user’s choice), zip them all together and save the resulting asset package. Note that while all assets have a Name, not all have a Long Name. If the user opts to rename the asset files using Long Name, Name should be used when Long Name is not defined.