A picture containing laser

Description automatically generated

|  |
| --- |
| BCX  Health & Retail Solutions  Functional & Technical Specification  Clinic & Unisolv Modernization |

**DOCUMENT VERSION 1.0**

**2021-08-24**

**ARCHITECT / CONTRIBUTORS / DEVELOPERS**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Department** |
| Douw Muller | Architect | BCX |

**TEAM**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Contact Information** |
|  |  |  |
|  |  |  |

**REVIEWERS**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Contact Information** |
| Douw Muller |  |  |

**VERSION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **New Functionality** | **Developer** |
| 2021-07-23 | 1.0.0 | Initial Draft | Douw Muller |

**Table of Contents**

[Introduction 5](#_Toc84404611)

[1.1 Terms/Acronyms & Definitions 5](#_Toc84404612)

[1. Data Bureau 6](#_Toc84404613)

[1.1 Overview 6](#_Toc84404614)

[1.2 Schema 6](#_Toc84404615)

[1.3 Product Price Index 9](#_Toc84404616)

[1.3.1 Overview 9](#_Toc84404617)

[1.3.2 Schema & Data Files 10](#_Toc84404618)

[1.3.3 Functionality 12](#_Toc84404619)

[1.4 Claim Admin Office 17](#_Toc84404620)

[1.4.1 Overview 17](#_Toc84404621)

[1.4.2 Schema & Data Files 17](#_Toc84404622)

[1.4.3 Functionality 17](#_Toc84404623)

[1.5 Medical Aid Option Information 19](#_Toc84404624)

[1.5.1 Overview 19](#_Toc84404625)

[1.5.2 Schema & Data Files 19](#_Toc84404626)

[1.5.3 Functionality 20](#_Toc84404627)

[1.6 Medical Aid Schemes 22](#_Toc84404628)

[1.6.1 Overview 22](#_Toc84404629)

[1.6.2 Schema & Data Files 22](#_Toc84404630)

[1.6.3 Functionality 23](#_Toc84404631)

[1.6.4 23](#_Toc84404632)

[1.7 Product Price Detail 26](#_Toc84404633)

[1.7.1 Overview 26](#_Toc84404634)

[1.7.2 Schema & Data Files 26](#_Toc84404635)

[1.7.3 Functionality 27](#_Toc84404636)

[1.8 Product Price Equivalence 29](#_Toc84404637)

[1.8.1 Overview 29](#_Toc84404638)

[1.8.2 Schema & Data Files 30](#_Toc84404639)

[1.8.3 Functionality 31](#_Toc84404640)

[1.9 Product Price Formulary 35](#_Toc84404641)

[1.9.1 Overview 35](#_Toc84404642)

[1.9.2 Schema & Data Files 35](#_Toc84404643)

[1.9.3 Functionality 36](#_Toc84404644)

[1.10 Translation Files – Active Ingredients 38](#_Toc84404645)

[1.10.1 Schema & Data Files 38](#_Toc84404646)

[1.10.2 Functionality 38](#_Toc84404647)

[1.11 Translation Files – Drug Types 40](#_Toc84404648)

[1.11.1 Schema & Data Files 40](#_Toc84404649)

[1.11.2 Functionality 41](#_Toc84404650)

[1.12 Translation Files – Suppliers 41](#_Toc84404651)

[1.12.1 Schema & Data Files 41](#_Toc84404652)

[1.12.2 Functionality 42](#_Toc84404653)

# Introduction

## Terms/Acronyms & Definitions

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definition** | **Description** |
|  |  |  |
|  |  |  |

# Data Bureau

## Overview

Data Bureau facilitates the import from Unisolv Data. This data will be maintained in the Data Bureau module and any updates will be sync back to Unisolv using backward compatibility files.

Data that will be maintained in Data Bureau:

* Product Price Index
* Product Price Index Details

Datasets will be differentiated by:

* Version

Indicate the version of the dataset. Any committed changes will cause a new version to be created.

* Tags

Tags can be linked to a dataset to indicate the purpose of the datasets. When datasets are sync to clients via Web-Hooks, Tags can be used to filter datasets that will be sync to specified client.

Unisolv files needed for the datasets imports are created with Migration:

Migrations > Tenants > DatabureauCreator.cs

* CreateTranslationDrugTypes
* CreateCommonFiles
  + Create rows in the File Repository that point to the import files.
* CreateTags

## Schema

**productindexes**

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | Id |  | Referenced By:  productformularygroups.ProductIndexId |
|  | Name | longtext | Name of Tag |

**productformularies**

(Note: not all columns are listed here)

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | DiscontDate | datetime(6) | Date when this formulary was discontinued |
|  | EffectDate | datetime(6) | Date when this formulary came into effect |
|  | GenericDrug | tinyint(1) |  |
|  | Group | int |  |
|  | **Id** | bigint |  |
|  | MedicalPrice |  |  |
|  | PackQty |  |  |
|  | Priority |  |  |
|  | ProductFormularyGroupId |  | FK: productformularygroups.Id |
|  | ProductIndexId |  | FK: productindexes.Id |
|  | SubGroup |  |  |
|  | Tags |  |  |
|  | UPD |  |  |

**productformularygroups**

(Note: not all columns are listed here)

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | Code |  |  |
|  | **Id** |  | Referenced by:  productformularies.ProductFormularyGroupId |
|  | Tags |  |  |
|  | TotalFormularies |  |  |

**productformularyprices**

(Note: not all columns are listed here)

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | DiscontDate |  | Date when this formulary price was discontinued |
|  | EffectDate |  | Date when this formulary price came into effect |
|  | GenericDrug | tinyint(1) |  |
|  | **Id** | bigint |  |
|  | MedicalPrice |  |  |
|  | Priority | Int |  |
|  | ProductFormularyId | bigint | FK: productformularies.Id |

**tags**

This table contains all tags that can be linked to datasets in the Data Bureau to enable clients to subscribe to specific tags only and therefore only get updates for the required datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | Id | char(36) |  |
|  | Name | longtext | Name of Tag |

**updatepriceversions**

Contains a list of all versions for the Data Bureau. The Data Bureau consists of a set of datasets and if any set is updated and committed that creates a new version. (Note: not all columns are listed here)

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | **Id** |  |  |
|  | TagId |  | Link to tags.Id |
|  | TotalChanges | Int | Indicate the total changes for this version across all datasets |
|  | TotalClaimAdmin | Int | Indicate changes for this version for the Claim Admin dataset |
|  | TotalMedicalAidOptions | int | Indicate changes for this version for the Medical Aid Options dataset |
|  | TotalMedicalAidSchemes | int | Indicate changes for this version for the Medical Aid Schemes dataset |
|  | TotalProductDetails | int | Indicate changes for this version for the Product Details dataset |
|  | TotalProductEquivalenceATC | int | Indicate changes for this version for the Product Equivalence ATC dataset |
|  | TotalProductEquivalenceGeneric | int | Indicate changes for this version for the Product Equivalence Generic dataset |
|  | TotalProductFormularies | int | Indicate changes for this version for the Product Formularies dataset |
|  | TotalProductIndex | int | Indicate changes for this version for the Product Index dataset |
|  | TotalTranslationActiveIngredients | int | Indicate changes for this version for the Translation Active Ingredients dataset |
|  | TotalTranslationDrugTypes | int | Indicate changes for this version for the Translation Drug Types dataset |
|  | TotalTranslationManufacturers | int | Indicate changes for this version for the Translation Manufacturer dataset |
|  | VersionNumber | int | Version Number |
|  | VersionStatus | int | 1:  0: |

**versionproductformulary**

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | ProductFormularyId |  | FK: productformularies.Id |
|  | UpdatePriceVersionId |  | FK: updatepriceversions.Id |

## Product Price Index

### Overview

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Navigation | Open the Cortex Management Service   * Login to ‘Default’ tenant as ‘admin’ * Select Data Bureau > Product Price Index > Products > Index |

### Schema & Data Files

#### Schema

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **productindexes** | Main entity | Core\DataBureau\Products\ProductIndex.cs |
| productformularies | One to Many | Core\DataBureau\Products\ProductFormulary.cs |
| productdetails | One to Many | Core\DataBureau\Products\ProductDetail.cs |
| translationmanufacturers | Many to One (FK) | Core\DataBureau\Products\Translations\TranslationManufacturer.cs |
| translationdrugtypes | Many to One (FK) | Core\DataBureau\Products\Translations\TranslationDrugType.cs |
| productequivalencegenerics | Many to One | Core\DataBureau\Products\ProductEquivalenceGeneric.cs |
| productequivalenceatcs | Many to One | Core\DataBureau\Products\ProductEquivalenceATC.cs |

**tags**

This table contains all tags that can be linked to datasets in the Data Bureau to enable clients to subscribe to specific tags only and therefore only get updates for the required datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | Id | char(36) |  |
|  | Name | longtext | Name of Tag |

**updatepriceversions**

Contains a list of all versions for the Data Bureau. The Data Bureau consists of a set of datasets and if any set is updated and committed that creates a new version. (Note: not all columns are listed here)

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Name | Data Type | Description |
|  | TagId |  | Link to tags.Id |
|  | TotalChanges | Int | Indicate the total changes for this version across all datasets |
|  | TotalClaimAdmin | Int | Indicate changes for this version for the Claim Admin dataset |
|  | TotalMedicalAidOptions | int | Indicate changes for this version for the Medical Aid Options dataset |
|  | TotalMedicalAidSchemes | int | Indicate changes for this version for the Medical Aid Schemes dataset |
|  | TotalProductDetails | int | Indicate changes for this version for the Product Details dataset |
|  | TotalProductEquivalenceATC | int | Indicate changes for this version for the Product Equivalence ATC dataset |
|  | TotalProductEquivalenceGeneric | int | Indicate changes for this version for the Product Equivalence Generic dataset |
|  | TotalProductFormularies | int | Indicate changes for this version for the Product Formularies dataset |
|  | TotalProductIndex | int | Indicate changes for this version for the Product Index dataset |
|  | TotalTranslationActiveIngredients | int | Indicate changes for this version for the Translation Active Ingredients dataset |
|  | TotalTranslationDrugTypes | int | Indicate changes for this version for the Translation Drug Types dataset |
|  | TotalTranslationManufacturers | int | Indicate changes for this version for the Translation Manufacturer dataset |
|  | VersionNumber | int | Version Number |
|  | VersionStatus | int | 1:  0: |

#### Data Files

Web.Mvc\wwwroot\File-Repository\Unisolv\Data\PHMDRG

Web.Mvc\wwwroot\File-Repository\Unisolv\Data\PHMDRG.vix

Web.Mvc\wwwroot\File-Repository\Unisolv\Data\phmdrg.xfd

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml |
| **Script** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml   * KTDatatable: url: 'GetProducts', |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetProducts   This method facilitates functionality (paging, sorting & filtering) of the KT table used and then calls the App Service method |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetProductIndexes   Return product indexes. Also include foreign key relationship data: translationdrugtypes & translationmanufacturers |

#### Sub Lists

GetProductDetailPrices

GetProductFormularyPrices

GetProductFormulariesByNappi

#### List Page (Versions)

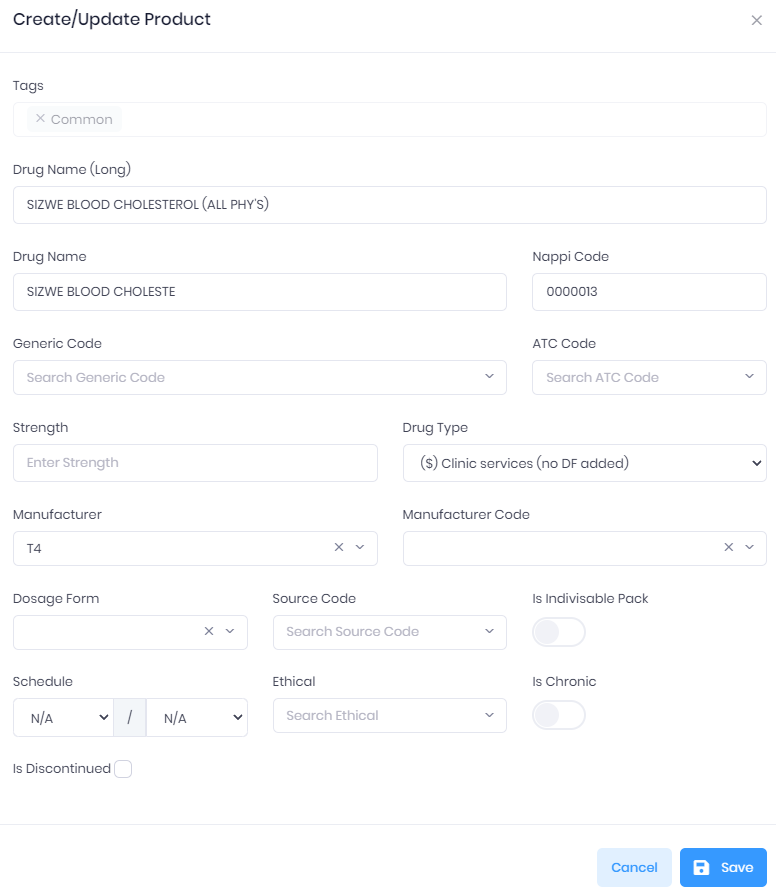
Datasets are linked to a version and a tag. Tags are stored in table ‘tags’. Versions are stored in table ‘updatepriceversion’

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * var tagsDataTable = \_$tagsTable.KTDatatable({ |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersion   This method facilitates functionality (paging, sorting & filtering) of the KT table used and then calls the App Service method |
| **App Service** | Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersions   * This will return all the tags linked to a version. |

Expand Tag Version to see the changes for the selected version

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * content: subTableInit, |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges   This method facilitates functionality (paging, sorting & filtering) of the KT table used and then calls the App Service method |
| **App Service** | Version Changes:  Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes:  Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

Edit



|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml |
| **Script** | function btnEditIndex(id) { |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateProductIndexModal.cshtml |
| **Script** | Web.Mvc\wwwroot\  view-resources\Areas\App\Views\DataBureau\Products\\_CreateUpdateProductIndexModal.js |
| Generic Code | |
| **Script** | //Generic  \_modalManager.getModal().find('#ProductEquivalenceGenericId').select2({ |
| **App Service** | Application\DataBureau\DataBureauAppService.cs > GenericCodeList   * Return rows from entity ‘productequivalencegenerics’ |

Change History

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Products.cshtml   * button id="ImportButton"   The process status of this dataset is stored in table [abpsettings].  App.ProductPrice.Index  If the status is InProgress (1) the Import button will be disabled.  This setting will prevent crud functions to be performed while import process is busy.  You might have to restart the solution, as this status is stored in the zero framework app settings and this will not change if you just change the value in the database. |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_ProductIndexImportModel.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_ProductIndexImportModel.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems   Returns files stored in the file repository that include PHMDRG in the filename. |
| **Import**  Run Unisolv Runtime @ C:\Unisolv\888.bat before you do the import. | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportProductIndex   Check if vix file exists for this import.  Call background job to start import. |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute * Set the file id and executes ‘UpdateStatus’, ‘CreateXFDUnisolvOutput’ and another background job ‘ImportData’   Note:  If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).   * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData * If the 'TranslationDrugTypeText' have a value, but 'TranslationDrugTypeId' is empty, a new entity must be added to the TranslationDrugType repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so father updates will not cause a new version again. * If the 'TranslationManufacturerText' have a value, but 'TranslationManufacturerId' is empty, a new entity must be added to the TranslationManufacturer repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so further updates will not cause a new version again. |

#### Create/Update

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateProductIndexModel.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_CreateUpdateProductIndexModel.js |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 8,9 |

## Claim Admin Office

### Overview

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Dev Ops User Story | 1262 |

### Schema & Data Files

#### Data Files

\wwwroot\File-Repository\Unisolv\Data\PHMCEN

\wwwroot\File-Repository\Unisolv\Data\PHMCEN.vix

\wwwroot\File-Repository\Unisolv\Data\phmcen.xfd

This is the data files from Unisolv.

### Functionality

#### Import Data

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_ClaimAdminImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_ClaimAdminImportModal.js |
| **File Lookup**  **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs > ImportClaimAdmin |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs > Execute  - If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  - If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).  Application\BackgroundJobs\Imports\XFDJob.cs > UpdateStatus  Application\BackgroundJobs\Imports\XFDJob.cs > CreateXFDUnisolvOutput  Application\DataBureau\ImportAppService.cs > ImportData |

#### Create

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateClaimAdminModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Admin\  \_CreateUpdateClaimAdminModal.js |

#### Filter Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml   * div id="filterModal" |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Admin\  \_CreateUpdateClaimAdminModal.js |

#### List

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetClaimAdministrators |
| **App**  **Service** | Application\DataBureau\DataBureauAppService.cs   * GetClaimAdmins |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 24,25 |

#### Versioning

|  |  |
| --- | --- |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * subTableInit |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
| **App Service** | Version Changes:  Cortex.Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes:  Cortex.Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

#### Processes

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Processes.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
|  | .Core\Configuration\AppSettings.cs |

## Medical Aid Option Information

### Overview

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Dev Ops User Story | 1117 |
|  |  |

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
| medicaloptions |  |
| medicaloptioncosts |  |
| medicaloptionmessages |  |
| medicaloptionschedules |  |
| medicaloptionconfiguration |  |
| versionmedicaloption |  |
| updatepriceversions |  |

#### Data Files

\wwwroot\File-Repository\Unisolv\Data\PHMCEN

\wwwroot\File-Repository\Unisolv\Data\PHMCEN.vix

\wwwroot\File-Repository\Unisolv\Data\phmcen.xfd

|  |  |
| --- | --- |
| Column | Description |
| MD-PAY-OFFICE | claimadmins.Code |
| ME-SCH | medicalaidschemes.SchemeCode |
| ME-CODE | OptionCode |

When importing Medical Aid Options, you should link the medicalaidoptions.MedicalAidSchemeId by using the ME-SCH and ME-CODE columns.

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\MedicalAidOptions.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetMedicalAidOptions |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetMedicalSchemeOptions |
| **Dto** | Application.Shared\DataBureau\Dto\MedicalAid   * MedicalAidOptionsListDto.cs * MedicalAidOptionsLookupRequest.cs |
| **Schema** | medicaloptions  medicalaidscheme |

#### List Page (Versions)

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_MedicalAidOptionImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\\_MedicalAidOptionImportModal.js |
| **File Lookup Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems |
| Import | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportMedicalAidOption |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs > Execute  - If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  - If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).  Application\BackgroundJobs\Imports\XFDJob.cs > UpdateStatus  Application\BackgroundJobs\Imports\XFDJob.cs > CreateXFDUnisolvOutput  Application\DataBureau\ImportAppService.cs > ImportData  Application.Shared\DataBureau\Base\Entities\MedicalAidOptionConverterDto.cs |

Note: Run the Unisolv runtime at C:\Unisolv\888.bat

#### Create

|  |  |
| --- | --- |
| **Modal**  **View** |  |
| **Script** |  |

#### Filter Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml   * div id="filterModal" |
| **Script** |  |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 24,25 |

#### Versioning

|  |  |
| --- | --- |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * subTableInit |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
| **App Service** | Version Changes:  Cortex.Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes:  Cortex.Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

## Medical Aid Schemes

### Overview

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Dev Ops User Story | 1117 |

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
|  |  |

#### Data Files

\wwwroot\File-Repository\Unisolv\Data\PHMCEN

\wwwroot\File-Repository\Unisolv\Data\PHMCEN.vix

\wwwroot\File-Repository\Unisolv\Data\phmcen.xfd

### Functionality

### 

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\MedicalAids.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetMedicalAidSchemes |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetMedicalSchemes |
| **Dto** | Application.Shared\DataBureau\Dto\MedicalAid   * MedicalAidOptionsListDto.cs * MedicalAidOptionsLookupRequest.cs |
| **Schema** | medicaloptions  medicalaidscheme |

#### List Page (Versions)

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_MedicalAidSchemeImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_MedicalAidSchemeImportModal.js |
| **File Lookup Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems |
| **Import**  Run Unisolv Runtime @ C:\Unisolv\888.bat before you do the import. | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportMedicalAidScheme |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute * UpdateStatus * CreateXFDUnisolvOutput   Application\DataBureau\ImportAppService.cs   * ImportData   Application.Shared\DataBureau\Base\Entities\MedicalAidOptionConverterDto.cs |

#### Create

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateMedicalAidSchemeModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\MedicalAid\   * \_CreateUpdateMedicalAidSchemeModal.js |

#### Filter Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml   * div id="filterModal" |
| **Script** |  |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 20,21,22,23 |

#### Versioning

|  |  |
| --- | --- |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * subTableInit |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
| **App Service** | Version Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

#### Discontinue

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\MedicalAids.cshtml   * btnDiscontIndex |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * DiscontinueMedicalAidScheme |

#### Restore

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\MedicalAids.cshtml   * btnRestoreDiscontIndex |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * DiscontinueRestoreMedicalAidScheme |

#### Save (Create & Edit)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\MedicalAids.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\MedicalAid\   * \_CreateUpdateMedicalAidSchemeModal.js * saveMedicalAidScheme |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * SaveMedicalAidScheme |
| **Job** | Job  Application\BackgroundJobs\Imports\MedicalAidSchemeUpdateJob.cs  Arguments  Application.Shared\BackgroundJobs\Args\Imports\MedicalAidSchemeUpdateJobArgs.cs |
| **App Service** | Application\DataBureau\ImportAppService.cs > UpdateMedicalAidScheme |

#### Processes

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Processes.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * Processes   Core\Configuration\AppSettings.cs |

#### Webhook Integration

#### Backward Compatibility File (Unisolv)

## Product Price Detail

### Overview

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Dev Ops User Story |  |

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
| productdetails |  |
| productdetailprices |  |

#### Data Files

…\wwwroot\File-Repository\Unisolv\Data\PHMDSP

…\wwwroot\File-Repository\Unisolv\Data\PHMDSP.vix

…\wwwroot\File-Repository\Unisolv\Data\phmdsp.xfd

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ProductDetails.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetProductDetails * GetProductDetailPrices |

#### List Page (Versions)

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_ProductDetailImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_ProductDetailImportModal.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems |
| **Import**  Run Unisolv Runtime @ C:\Unisolv\888.bat before you do the import. | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportProductDetail |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData |

#### Create/Update

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateProductDetailModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products \   * \_CreateUpdateProductDetailModal.js |

#### Filter Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml   * div id="filterModal" |
| **Script** |  |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 10,11 |

#### Versioning

|  |  |
| --- | --- |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * subTableInit |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
| **App Service** | Version Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

#### Processes

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Processes.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * Processes   Core\Configuration\AppSettings.cs |

#### Webhook Integration

#### Backward Compatibility File (Unisolv)

## Product Price Equivalence

### Overview

Pharmaceutical Equivalents Drug products in identical dosage forms that contain the same active ingredient (s), ie, the same salt or ester, are of the same dosage form, use the same route of administration, and are identical in strength or concentration can be classified as Product Equivalences.

The WHO assigns **ATC** codes to all active substances contained in **medicines** based on the therapeutic indication for the medicine. Using the **ATC** code, active substances are classified in groups at five different levels according to the organ or system on which they act and their therapeutic, **pharmacological** and chemical properties.

|  |  |
| --- | --- |
| Solution | Cortex Management Service |
| Navigation | Open the Cortex Management Service   * Login to ‘Default’ tenant as ‘admin’ * Select Data Bureau > Product Price Index > Products > Equivalence |

### Schema & Data Files

#### Schema

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| productequivalenceatcs |  |  |
| productequivalencegenerics | Core\DataBureau\Products\ProductEquivalenceGeneric.cs |  |
| productequivalencegenerictranslationactiveingredient | Core\DataBureau\Products\ProductEquivalenceGenericTranslationActiveIngredient.cs |  |

**productequivalenceatc**

This entity contains the active substances contained in the applicable medicine.

**productequivalencegenerics**

This entity contains generic/equivalent products to the products stored in the productindexes entity.

**productequivalencegenerictranslationactiveingredient**

This entity contains active ingredients that is found in a generic/equivalent product.

#### Data Files

…\wwwroot\File-Repository\Unisolv\Data\PHMGEN

…\wwwroot\File-Repository\Unisolv\Data\PHMGEN.vix

…\wwwroot\File-Repository\Unisolv\Data\phmgen.xfd

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ProductEquivalences.cshtml |
| **Script** | url: 'GetProductEquivalenceGenerics', |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetProductEquivalenceGenerics |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetProductDetails * GetProductDetailPrices |

#### List Page (Versions)

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ProductEquivalences.cshtml   * button id="ImportButton"   The process status of this dataset is stored in table [abpsettings]  If the status is InProgress the Import button will be disabled.  This setting will prevent crud functions to be performed while import process is busy.  You might have to restart the solution, as this status is stored in the zero framework app settings and this will not change if you just change the value in the database. |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_ProductEquivalenceImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_ProductEquivalenceImportModal.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems   Returns files stored in the file repository that include **PHMGEN** in the filename. |
| **Import** | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportProductEquivalence   Check if vix file exists for this import.  Call background job to start import. |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute * Set the file id and executes ‘UpdateStatus’, ‘CreateXFDUnisolvOutput’ and another background job ‘ImportData’   Note:  If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).   * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData |

#### Create/Update

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateProductDetailModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products \   * \_CreateUpdateProductDetailModal.js |

#### Filter Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ClaimAdmins.cshtml   * div id="filterModal" |
| **Script** |  |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 10,11 |

#### Versioning

|  |  |
| --- | --- |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * subTableInit |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges |
| **App Service** | Version Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes  Cortex.Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

#### Processes

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Processes.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * Processes   Core\Configuration\AppSettings.cs |

#### Webhook Integration

#### Backward Compatibility File (Unisolv)

## Product Price Formulary

### Overview

A formulary is a **continually updated list of medications and related information**, representing the clinical judgment of pharmacists, physicians, and other experts in the diagnosis and/or treatment of disease and promotion of health. A written, unbiased evaluation of a specific medication.

### Schema & Data Files

#### Schema

Diagram

Description automatically generated

#### Data Files

…\wwwroot\File-Repository\Unisolv\Data\PHMMAF

…\wwwroot\File-Repository\Unisolv\Data\PHMMAF.vix

…\wwwroot\File-Repository\Unisolv\Data\phmmaf.xfd

### Functionality

#### List Page (Data)

The main table returns the Formulary Groups. Inside the main table there is a sub table for Formularies and inside this table a sub table for Formulary Details (Prices)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ProductFormulary.cshtml |
| **Script** | Web.Mvc\Areas\App\Views\DataBureau\ProductFormulary.cshtml   * Main Table: url: 'GetProductFormularyGroups', |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetProductFormularyGroups   This method facilitates the functionality (paging, sorting & filtering) of the KT table used and invokes the App Service method. |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetProductFormularyGroups   Returns formulary groups |

#### List Page (Versions)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ProductFormulary.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * var tagsDataTable = \_$tagsTable.KTDatatable({ |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersion   This method facilitates the functionality (paging, sorting & filtering) of the KT table and invokes the App Service method. |
| **App Service** | Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersions   * This will return all the tags linked to a version. |

Expand Tag Version to see the changes for the selected version

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ ProductFormulary.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Versions\Index.js   * content: subTableInit, |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetPriceVersionChanges   This method facilitates functionality (paging, sorting & filtering) of the KT table used and then calls the App Service method |
| **App Service** | Version Changes:  Application\DataBureau\DataBureauAppService.cs > GetTagPriceVersionChanges  Entity Changes:  Application\DataBureau\DataBureauAppService.cs > GetPriceEntityChanges |

Edit

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_ProductFormularyImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_ProductFormularyImportModall.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems   Returns files stored in the file repository that include **PHMMAF** in the filename. |
| **Import** | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportProductFormulary |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 18,19 |

## Translation Files – Active Ingredients

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
| translationactiveingredients |  |

Dependencies

|  |  |
| --- | --- |
| translationactiveingredients.Id | Get referenced by:  productequivalencegenerictranslationactiveingredient.TranslationActiveIngredientId |

#### Data Files

…\wwwroot\File-Repository\Unisolv\Data\PHMACT

…\wwwroot\File-Repository\Unisolv\Data\PHMACT.vix

…\wwwroot\File-Repository\Unisolv\Data\phmact.xfd

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\ActiveIngredients.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetTranslationActiveIngredients |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 12,13 |

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_TranslationActiveIngredientImportModal.cshtml |
| **Script** | .Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_TranslationActiveIngredientImportModal.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems |
| **Import**  Run Unisolv Runtime @ C:\Unisolv\888.bat before you do the import. | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportProductIndex |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute   If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).   * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData * If the 'TranslationDrugTypeText' have a value, but 'TranslationDrugTypeId' is empty, a new entity must be added to the TranslationDrugType repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so father updates will not cause a new version again. * If the 'TranslationManufacturerText' have a value, but 'TranslationManufacturerId' is empty, a new entity must be added to the TranslationManufacturer repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so further updates will not cause a new version again. |

## Translation Files – Drug Types

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
| translationdrugtypes |  |

Dependencies

|  |  |
| --- | --- |
| translationdrugtypes.Id | productequivalencegenerics.TranslationDrugTypeId references it  versiontranslationdrugtype.TranslationDrugTypeId references it |

Data gets seeded in .EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs > CreateTranslationDrugTypes

Table translationdrugtypes gets populated with the drug type values.

Table versiontranslationdrugtypes gets populated with the link between the drug type and the version.

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\DrugTypes.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetTranslationDrugTypes |

#### Create/Update

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateTranslationDrugTypeModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_CreateUpdateTranslationDrugTypeModal.js |

## Translation Files – Suppliers

### Schema & Data Files

#### Schema

|  |  |
| --- | --- |
| translationmanufacturer |  |

Dependencies

|  |  |
| --- | --- |
|  |  |

#### Data Files

…\wwwroot\File-Repository\Unisolv\Data\PHMSPL

…\wwwroot\File-Repository\Unisolv\Data\PHMSPL.vix

…\wwwroot\File-Repository\Unisolv\Data\phmspl.xfd

### Functionality

#### List Page (Data)

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\Manufacturers.cshtml |
| **Controller** | Web.Mvc\Areas\App\Controllers\DataBureauController.cs   * GetTranslationManufacturers |

#### Import Data

|  |  |
| --- | --- |
| **View** | Web.Mvc\Areas\App\Views\DataBureau\\_TranslationManufacturerImportModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Imports\   * \_TranslationManufacturerImportModal.js |
| **File Lookup** | |
| **Modal View** | Web.Mvc\Areas\App\Views\DataBureau\\_FileLookupModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\\_FileLookupModal.js |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * GetFileItems |
| **Import**  Run Unisolv Runtime @ C:\Unisolv\888.bat before you do the import. | |
| **App Service** | Application\DataBureau\DataBureauAppService.cs   * ImportTranslationManufacturer |
| **Job** | Application\BackgroundJobs\Imports\XFDJob.cs   * Execute   If you have a xfd file you can create a delimited output file during run-time (XFDJob.cs).  If you do not have a xfd file you can use the vtutil process to create an output file (VutilJob.cs).   * UpdateStatus   Update AppSettings.TenantProcesses.ProductPrice\_Index setting indicating that file is processing.  Stored in table [abpsettings]  This setting will prevent crud functions to be performed while import process is busy.   * CreateXFDUnisolvOutput   Set file properties.  Update bat file.  Create 'Data Output' folder (if applicable)  Create output file using Unisolv runtime    Application\DataBureau\ImportAppService.cs   * ImportData * If the 'TranslationDrugTypeText' have a value, but 'TranslationDrugTypeId' is empty, a new entity must be added to the TranslationDrugType repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so father updates will not cause a new version again. * If the 'TranslationManufacturerText' have a value, but 'TranslationManufacturerId' is empty, a new entity must be added to the TranslationManufacturer repo. * Also, a comparison needs to be done, to establish if this new added entity will cause the creation of a new version. * If the current version is in 'Draft' mode, this will not cause a new version, but if the version is 'Committed', this will cause the creation of a new version (the new version will default to 'Draft' status, so further updates will not cause a new version again. |

#### File Manager Service

File Manager entries will be created during the seed process when solution is started.

|  |  |
| --- | --- |
| **Seed** | EntityFrameworkCore\Migrations\Seed\Tenants\DataBureauCreator.cs   * CreateCommonFiles   Id = 14,15 |

#### Create/Update

|  |  |
| --- | --- |
| **Modal**  **View** | Web.Mvc\Areas\App\Views\DataBureau\\_CreateUpdateTranslationManufacturerModal.cshtml |
| **Script** | Web.Mvc\wwwroot\view-resources\Areas\App\Views\DataBureau\Products\Translations\   * \_CreateUpdateTranslationManufacturerModal.js |