ImportExportPF
Release Notes Vs. 1.2.1 07/25/2020

General consideration

The ImportExportPF app is aimed to move rules from development/test to production or between different DB. Today if a Rulestream user wants to move a change set, he must migrate the entire DB or re-implement in the production environment manually. The ImportExportPF app allows the user to select specific PartFamilies or all the PartFamilies belonging to a given Rulestream application and moving them to the destination environment. The ImportExportPF app is a standalone application reading from a source DB server and writing to destination DB server, without using API's and/or web services, but creating an intermediate neutral file in MindMap format. The advantages in using the Db to Db approach are:

- Minimal prerequisites. There is not even the need to have RS installed on the execution machine.
- RS application could be tracked into a CVS system. As the application creates a single neutral file with the translation of the RS application, this file could be placed into a CVS system to track modifications, releases etc.
- Documentation of RS application. As the neutral file is a mind map file readable by Mind Map graphic editors (for example FreePlane), this is an effective description of the application
- Performances. The DB to DB approach is probably the fastest way to perform the import/export operation
- RS Release independence. A RS application developed in RS 8.17.0 can be moved in a RS 8.16.x or 8.15.x environment
- No RS licenses are copied/overwritten/modified

Application Description

The ImportExportPF app has the following general features:

- External utility (not integrated into Architect)

- DB login into the Source and Destination DBs with SqlServer credentials or Windows Authentication
- The XML ini file created into the User's MyDocuments folder, while the log file is created in c:\temp or in c:\ProgramData\Rulestream\8.17.0\logs. If both locations do not exist or are not accessible no log file is created
- Rule model export/import
  - Selective export/import of Part Families (pick the ones you want)
    If a PF by that name already exists, the user the user will have the following options:
    - Update the PF
    - Delete and recreate the PF
    - Interrupt export procedure
  - Import/Export the entire model associated to an existing application
  - Additive import/export. In case of an application with multiple Top-Level Part Families the procedure will write as many neutral files as the number of Top-Level Part Families. With this option the user can select, read all these files and deploy in one shot all the Part Families into the Destination Db
  - Import "As Plug-in". In this case a branch of the source PCM is added to the destination system. Source and destination system are two separate and independent RS applications. This option is useful to copy part of an application to a different one copying also the GUI
  - Export Related DB component data. This option allows the user to extract all the component DB tables referred by DB specs in the application and scripting them in the output folder
  - Export Master Documents files. This option allows the user to get all MD files connected to Specs and copying them to the output folder
- Subscriptions are supported
- Multiple contexts are supported
- The following features are not supported
  - Copy of Orphaned Parts
  - NX Piping Spec

The procedure creates an intermediate file as Mind Map xml file. The conversion process is a 2 steps process: Rulestream to Mind Map, Mind Map to Rulestream controlled by a single executable. The Mind Map intermediate file can be used for debug and /or documentation purposes. Therefore, the entire hierarchical part families structure will be read from Rulestream source DB, stored in the XML intermediate file and deployed into the destination DB

Eventual Categories, Units not existing in the Destination Db are created by the App.

Eventual Connections, Subpart Collections involving the use of Part Families not existing into Destination System, are created but the reference to these foreign Part Families is ignored.

The procedure is written in C# with Visual Studio 2019, developed and tested with RS 8.15.2, 8.16.4, 8.17.0. The DBMS in use will be Microsoft SqlServer 2014 R2

For any information needed please contact

Neri Alamanni email: neri.alamanni@configuratori.it cell. +39 348 2617589

Marco Lotti email: marco.lotti@configuratori.it cell. +39 335 7571048

Installation Description

To install the application please use the installer supplied: ImportExportPF_Setup.exe. Before installing a new version it is strongly suggested to uninstall the previous one

Usage Description

The application is tested extensively but it is impossible test all the possible situations in terms of rules text or Database conditions, therefore a preliminary backup of the destination DB is strongly suggested.

As the neutral file is an XML file try to avoid the use of ">" or "<" in the comments of the rules or in the caption of the steps. This does not harm the import/export operation but could prevent the visualization of the MindMap

After the installation an icon should be created in the desktop (Fig. 1). A double click on it activates the application
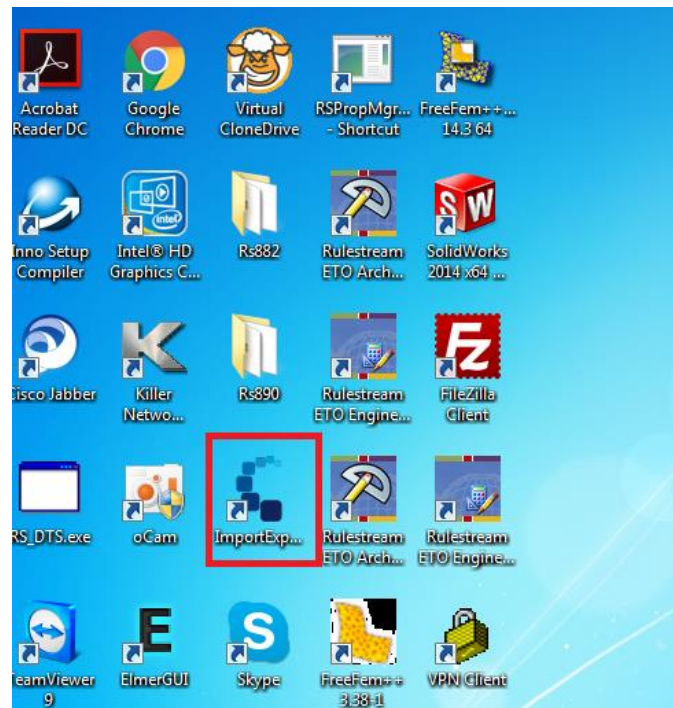
Fig. 1

The application GUI is displayed in Fig. 2. The user must perform the login into the source and/or destination databases. Of course, it is possible to perform the import/export operation at different times. In this case the user must perform the login into source database, to create the neutral file and then, in another time and, eventually on another pc, logging in the destination database, reading the neutral file and deploy the Rs App into the destination database.

The sequence of operations needed to export all/some Part Families belonging to a RS Application is (Fig.3):

1. Perform the login into the Source Server
2. Select the Source Database. Only the databases containing an "Application" table are displayed
3. Select the appropriate RS Application
4. Select the desired option to deal with the eventually existing Part Families in the Destination Database
5. Assign a name for the MindMap file to be written. By default, the name is the one of the Rs Application
6. Click on the button "Read From Source DB"

In Fig. 4 is shown the result of Export operation from RS. In the treeview control are displayed the corresponding Part Families, where the user can select which ones could be written into the neutral file. The Buttons "Select All", "Deselect All" select or deselect all items in the treeview.

The neutral file is a Mind Map file that could be displayed by some graphical presentation tool like FreePlane. The result can be seen in Fig. 5
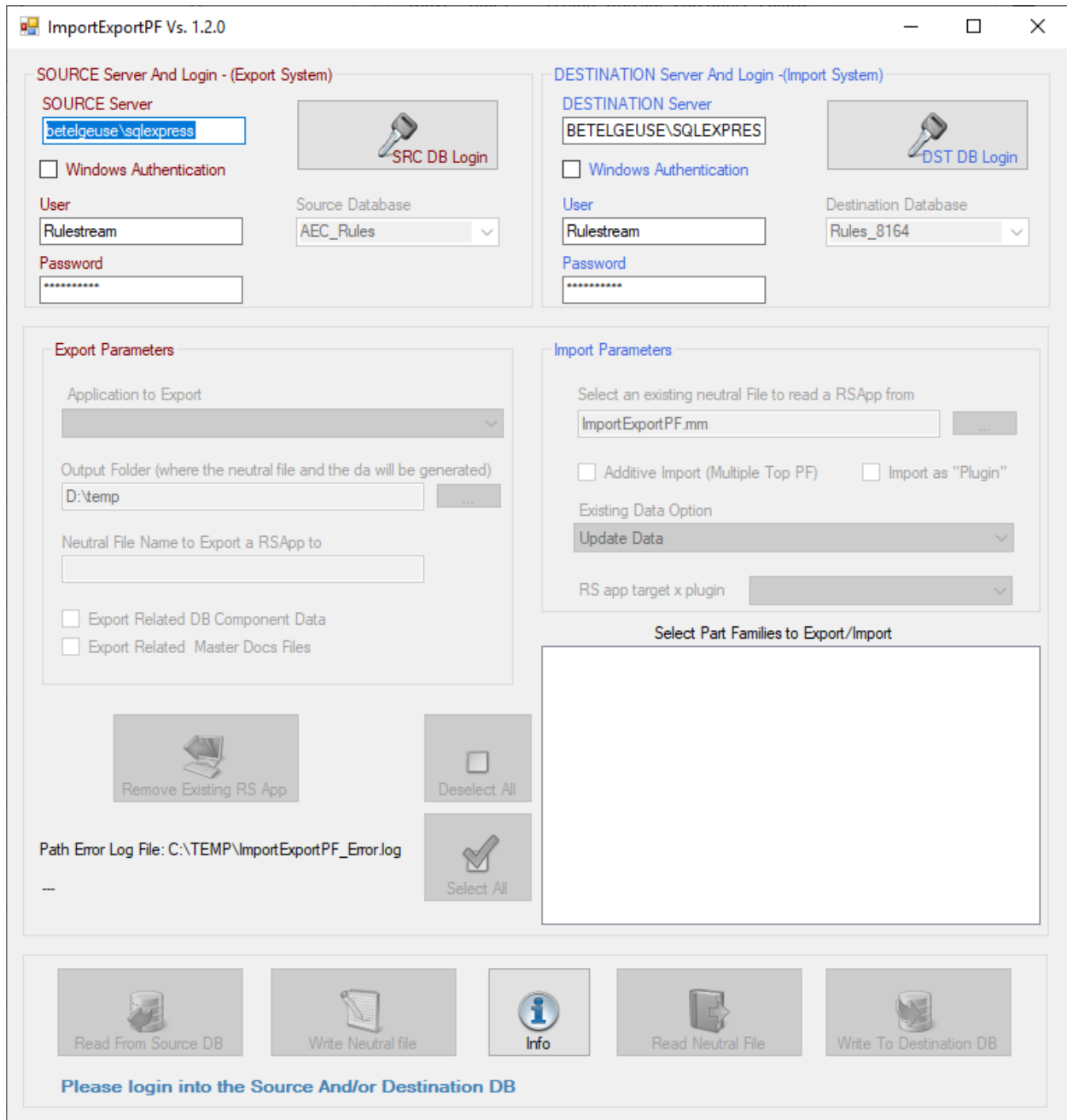
Fig. 2

The sequence of operations needed to import all/some Part Families from a neutral file to the destination DB is (Fig.6 & 7):

1. Perform the login into the Destination Server
2. Select the Destination Database
3. Select the desired option to deal with the eventually existing Part Families in the Destination Database
4. Select the neutral file to be read, using if needed the Ellipsis button

5. Click on the button "Read Neutral File"
6. Click on the button "Write to Destination DB"

The sequence "Read From Source DB", "Write Neutral File", "Read Neutral File", "Write To Destination DB" must be done even in case of a single session (i.e. writing and the reading the neutral file without closing the program)
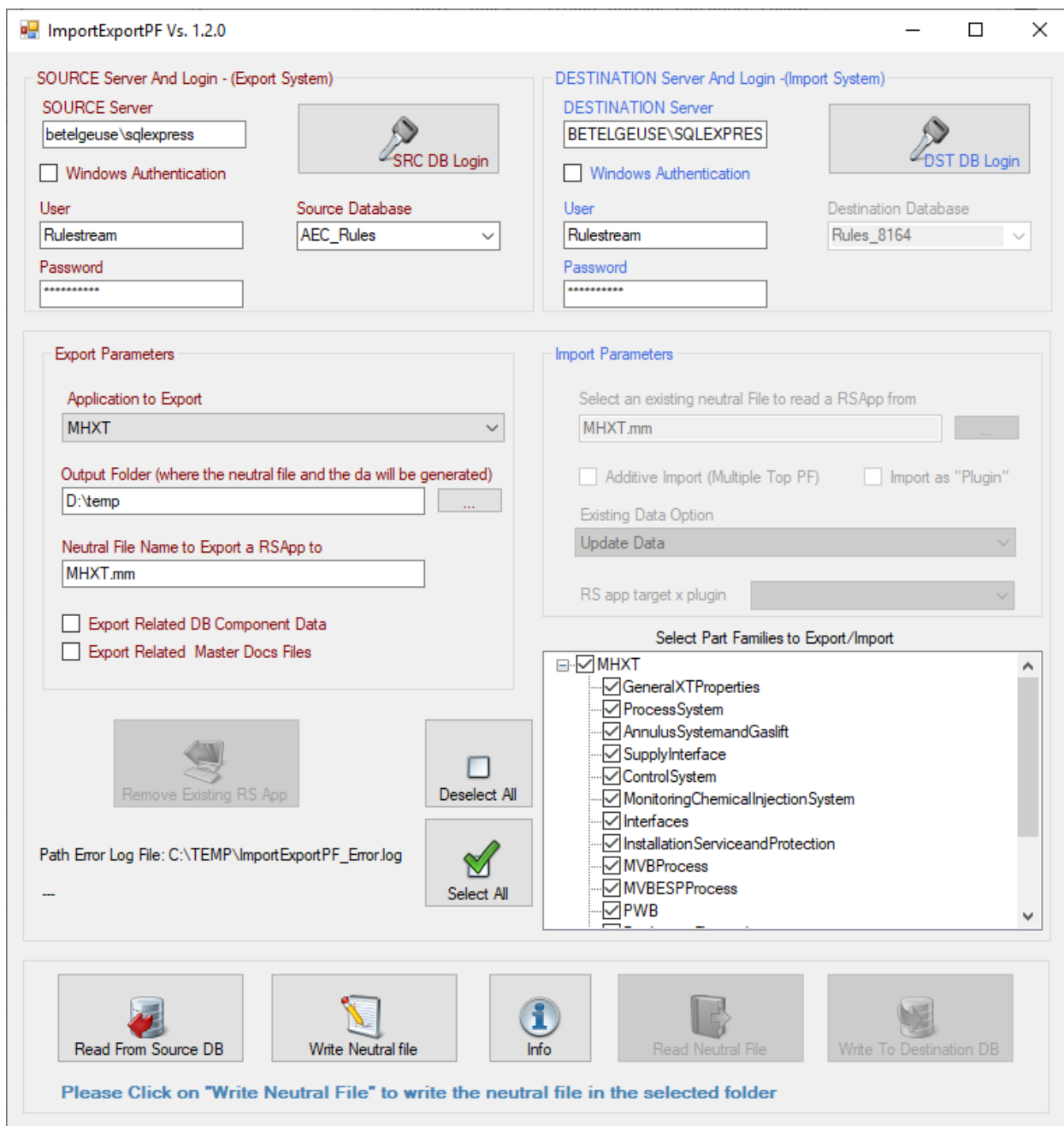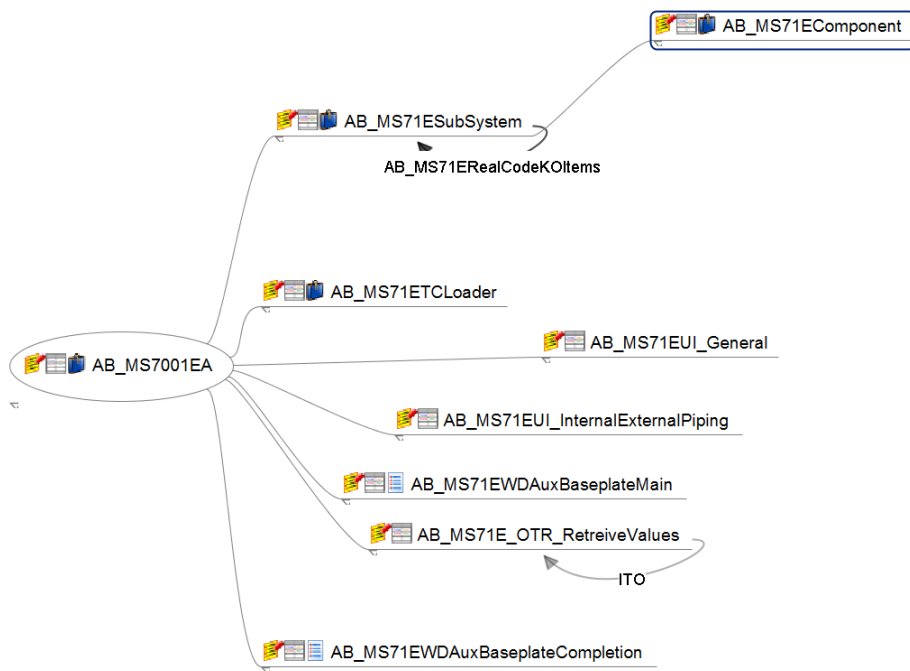

Fig. 3

Fig. 4

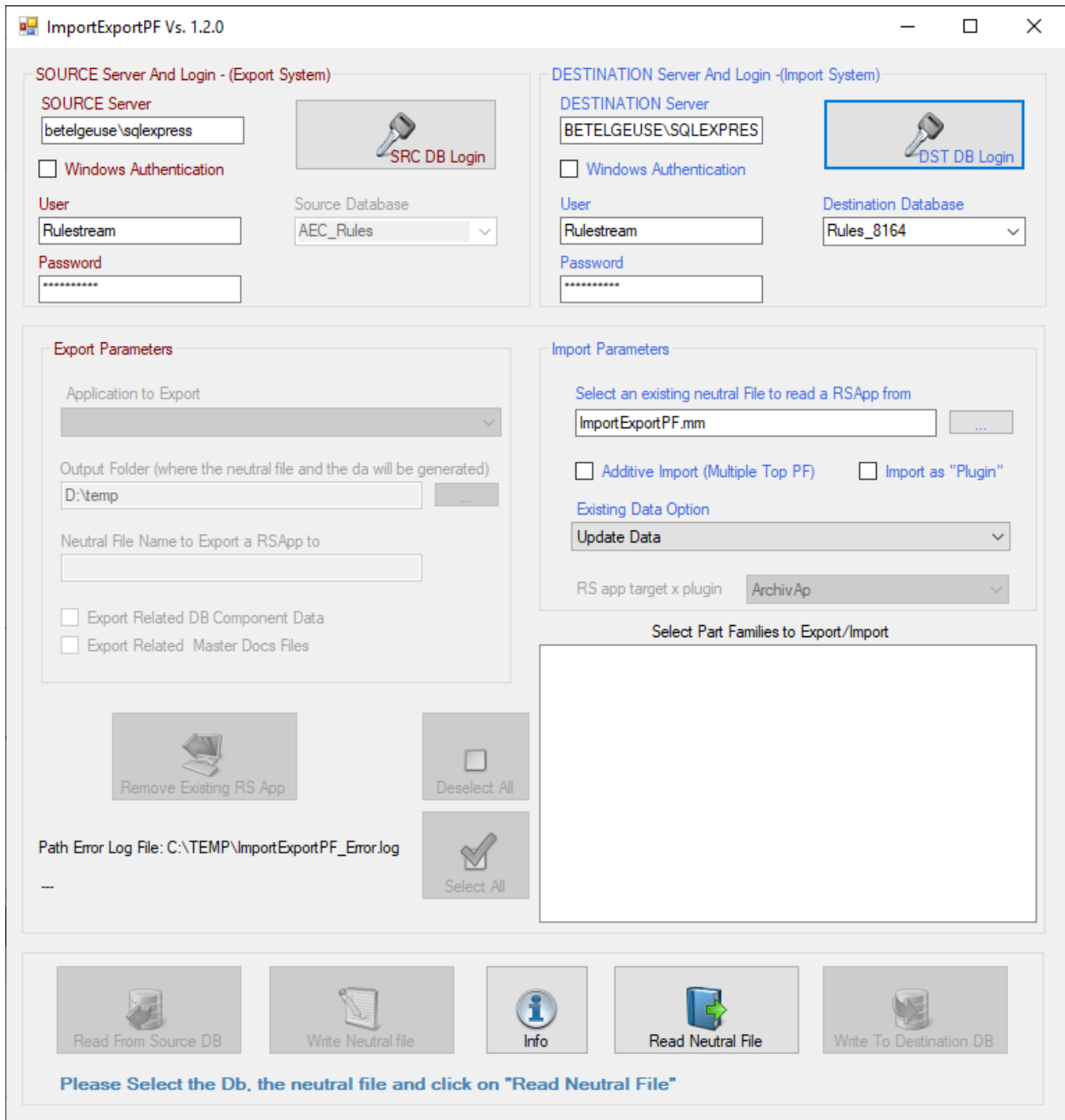AB_MS71EComponent

AB_MS71ESubSystem

AB_MS71ERealCodeKOItems

AB_MS71ETCLoader

AB_MS71EUI_General

AB_MS7001EA

AB_MS71EUI_InternalExternalPiping

AB_MS71EWDAuxBaseplateMain

AB_MS71E_OTR_RetreiveValues

ITO

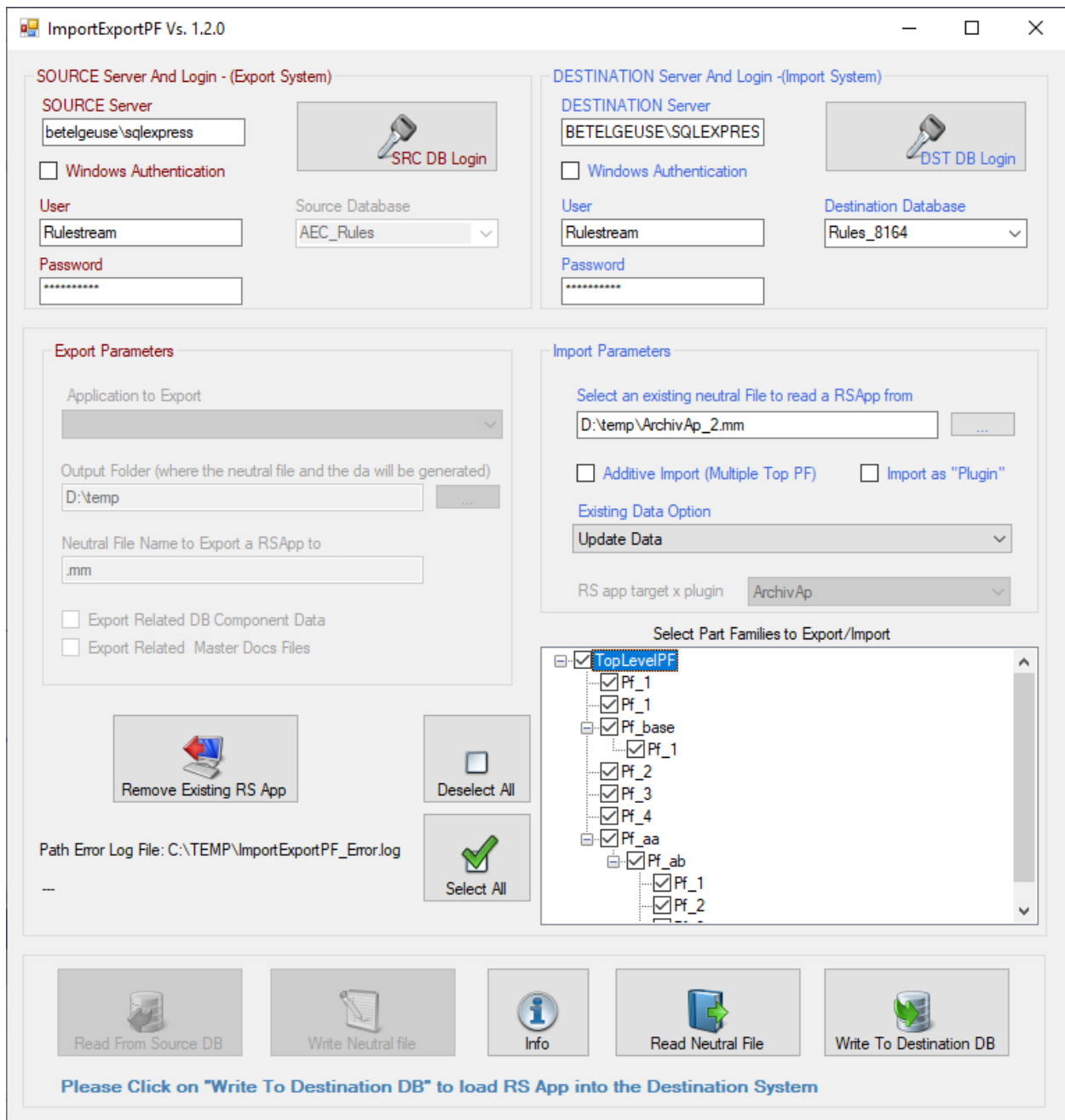AB_MS71EWDAuxBaseplateCompletion
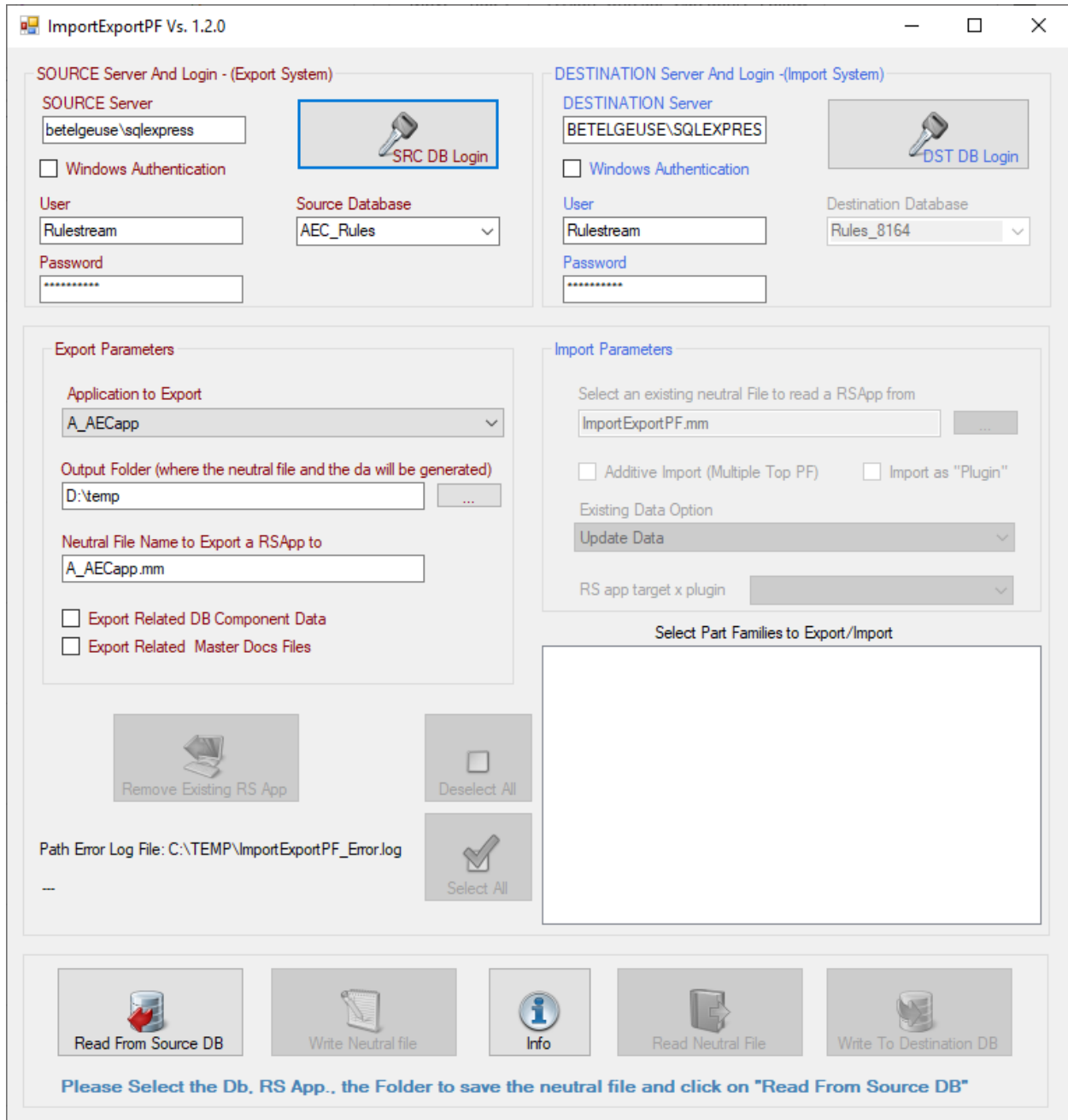
Fig. 5

Fig. 6

Fig. 7

In case of additive import (i.e. selecting more neutral files for an application with multiple Top-Level Part Family) the GUI is as displayed in Fig. 8. Once the Rs App is processed and the corresponding files are created the Warning Message in Fig. 9 is displayed.

To copy part of the PCM from an application to another you have to use the option "Import as Plugin". In this case the "Existing Data Option" must be set to "No Overwrite (adding missing entities only)" and the destination application must be set in "RS app target for plugin" (Fig.10)
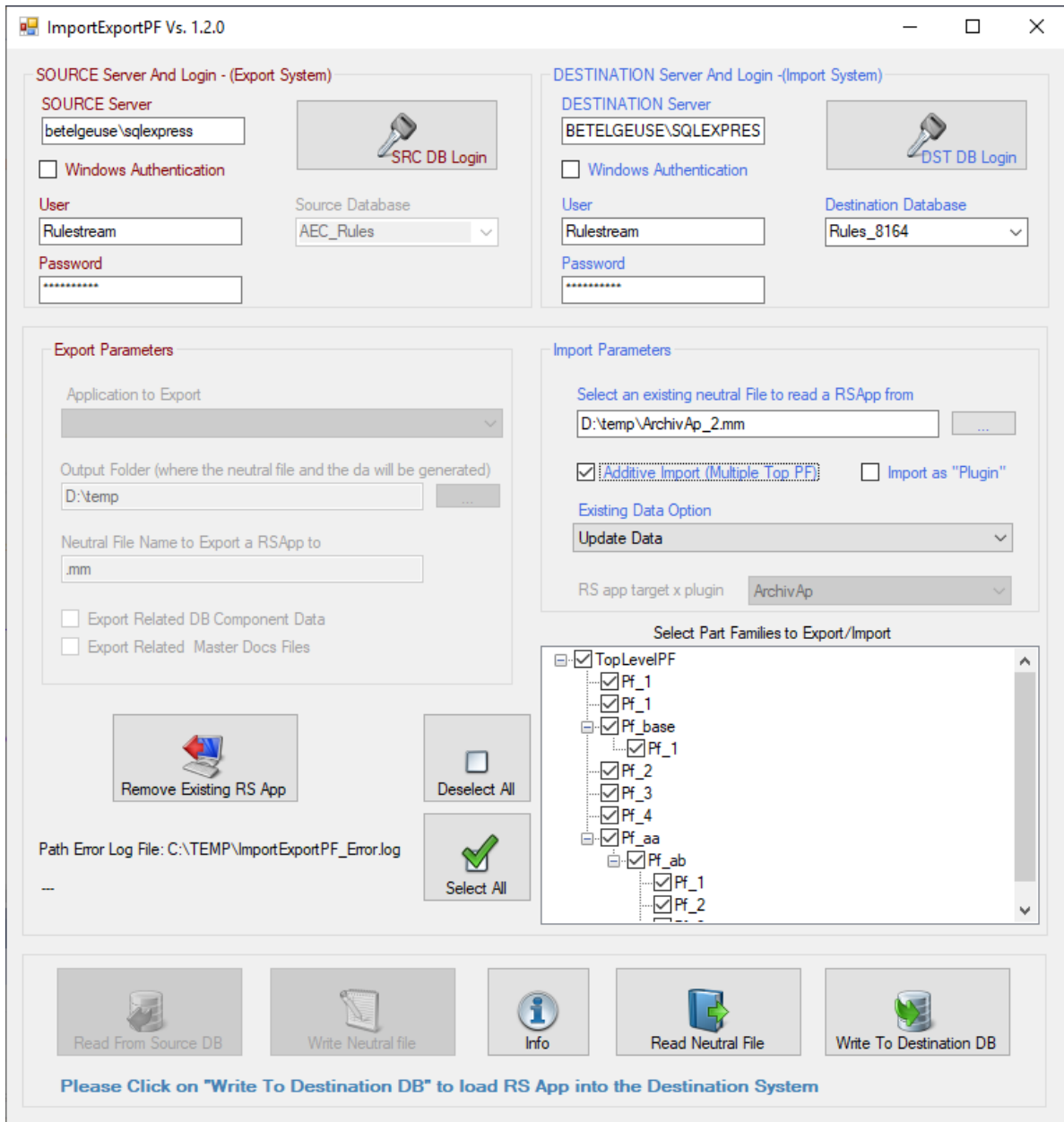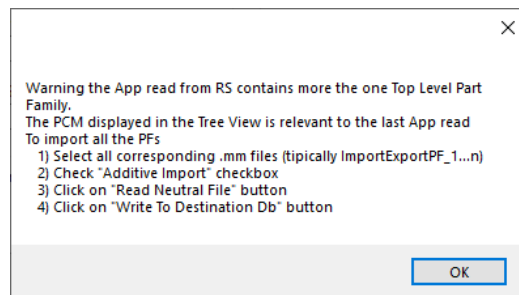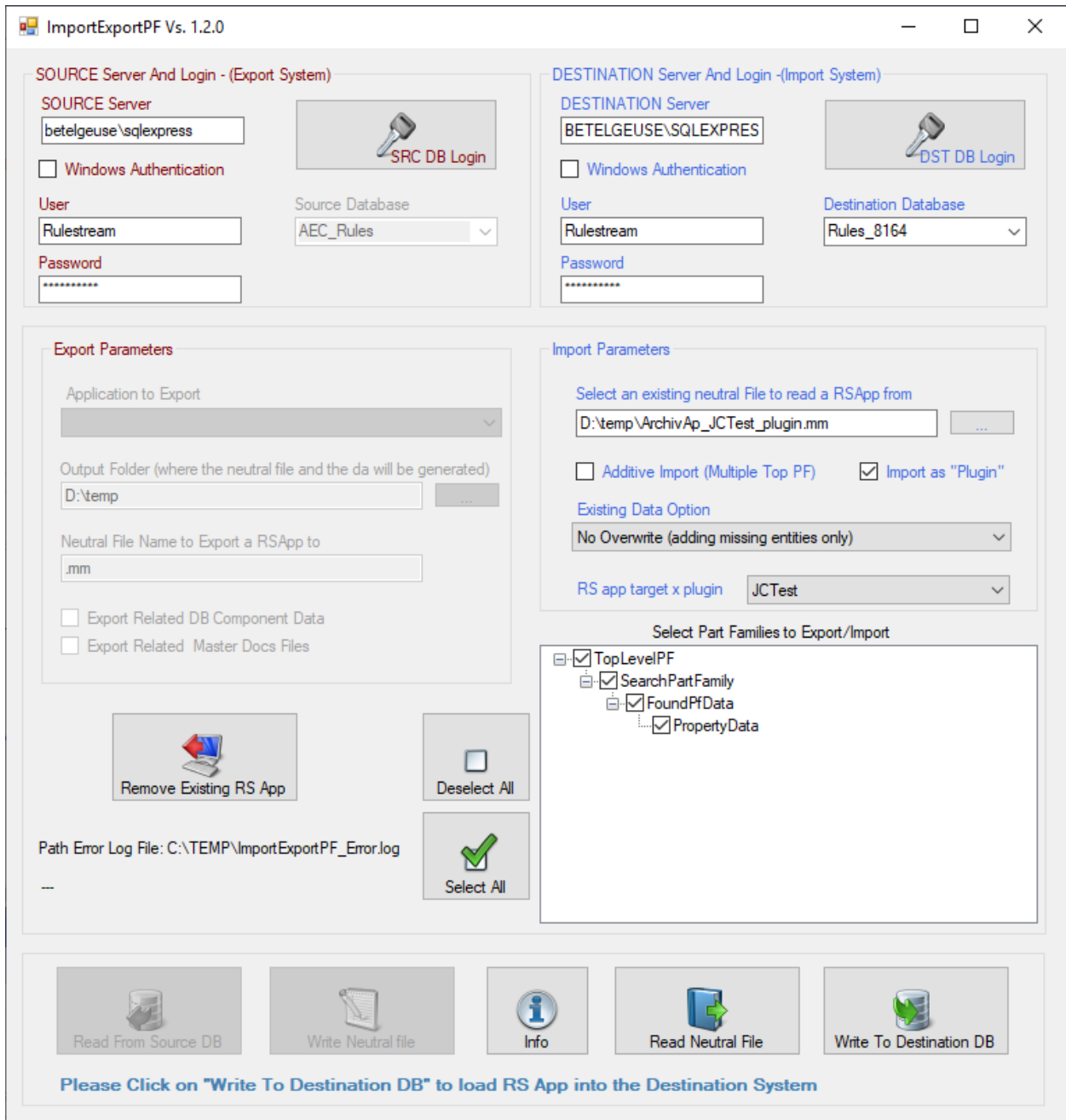
Fig. 8

Fig.9

Fig. 10

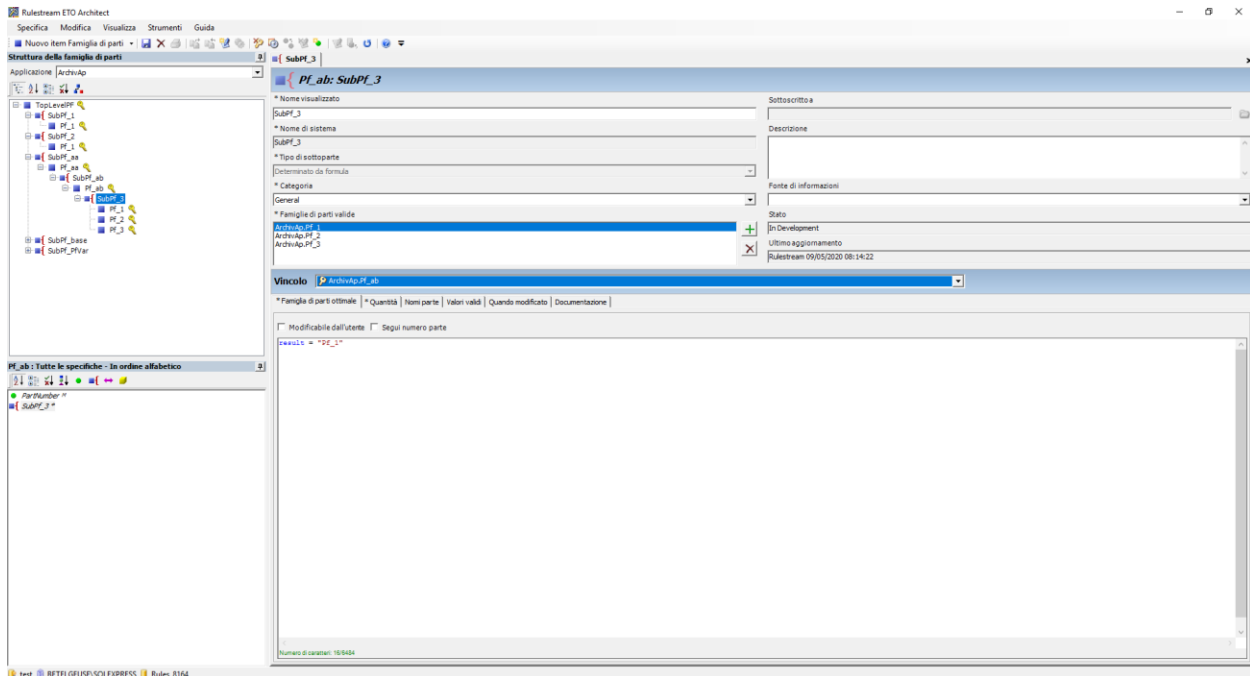In Fig. 11 can be seen the resulting application as imported in RS

Fig. 11

The tool can be used in three different ways to handle the existing Part Families in the destination system:

1. "Update Data" i.e. retaining the existing Part Families IDs.
2. "Delete and Recreate Data" i.e. deleting existing Part Families and creating new ones with, obviously, new IDs. This implies that any eventual Line Items previously created in Rs Engineer are no longer valid as they refer to Part Family IDs no longer existing. Trying to open existing Line Items after a "Delete and Recreate" will produce an error message as the one displayed in Fig.12
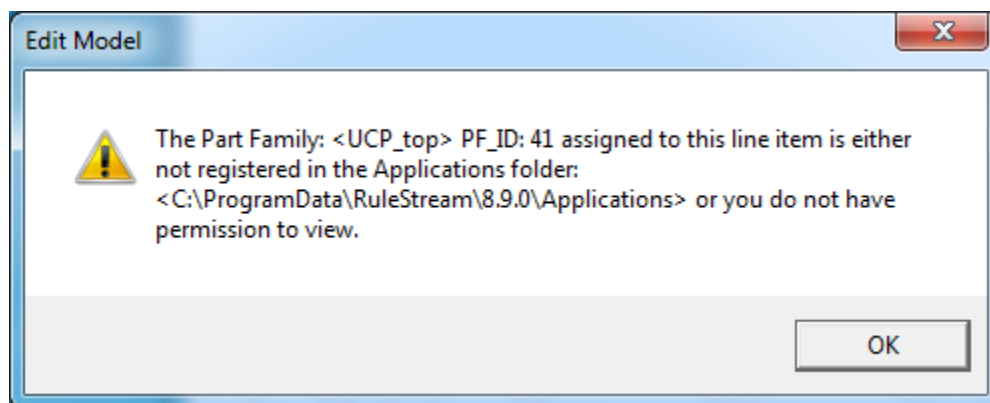3. "No Overwrite" only the missing entities in the destination system will be added

Fig. 12

The default approach for the Import Export tool is "Update Data" and selecting "Delete And Recreate" approach shows a warning message.

No orphaned parts will be involved in the import-export operation. The orphaned parts are Part Families not connected to the Top-Level Part Family, not belonging to any subpart, connection, subscription owned by the Top-Level Part Family. These orphaned parts are not detected and ignored during the export operations

No Formula History records are restored.

The removal the existing application deletes the RS App defined in the .mm file

The deletion of the eventually existing Rs application must be done with great care: this causes a forced deletion of all Part Families, Specs, Processes, Steps and of course all properties, Subparts, connections.

Any Line Items connected to the application will be lost as the relevant processes will be deleted

- Always perform a backup prior to perform the deletion
- Check the actual removal of the application with RS Architect and eventually repeat the deletion operation. In case of deletion of very large Rs Apps the timeout in execution of the stored procedures could truncate the operation. If so repeat the deletion until a complete removal of the app

The eventual presence of Offline Databases is now ignored