

.com Solutions Inc.



# **How to Import Visual FoxPro Applications into FmPro Migrator**

# How to Import Visual FoxPro Applications into FmPro Migrator

## 1 Step 1 - Get Info

- 1.1 Licensing FmPro Migrator 4
- 1.2 Visual FoxPro Conversion - Step 1 - Get Info 8

## 2 Step 2 - Create Table

- 2.1 Visual FoxPro Conversion - Step 2 - Create Table 19

## 3 Step 3 - Transfer Data

- 3.1 Visual FoxPro Conversion - Step 3 - Transfer Data 24

## 4 Import MetaData

- 4.1 Visual FoxPro Conversion - Import MetaData 32

## 5 Form/Report Conversion

- 5.1 Converting Forms/Reports and Layouts - GUI Tab 41

## 6 Code Conversion with the VFP Code Conversion Workbench

- 6.1 Using the Code Conversion Workbench 44
- 6.2 Using the VFP Code Conversion Workbench 56

# Step 1 - Get Info

## Licensing FmPro Migrator

---

As of FmPro Migrator v11, all FmPro Migrator downloads are fully functional with features unlocked via a single license key.

This section of the manual shows how to enter the license key to unlock the features within FmPro Migrator.

FmPro Migrator 11.01

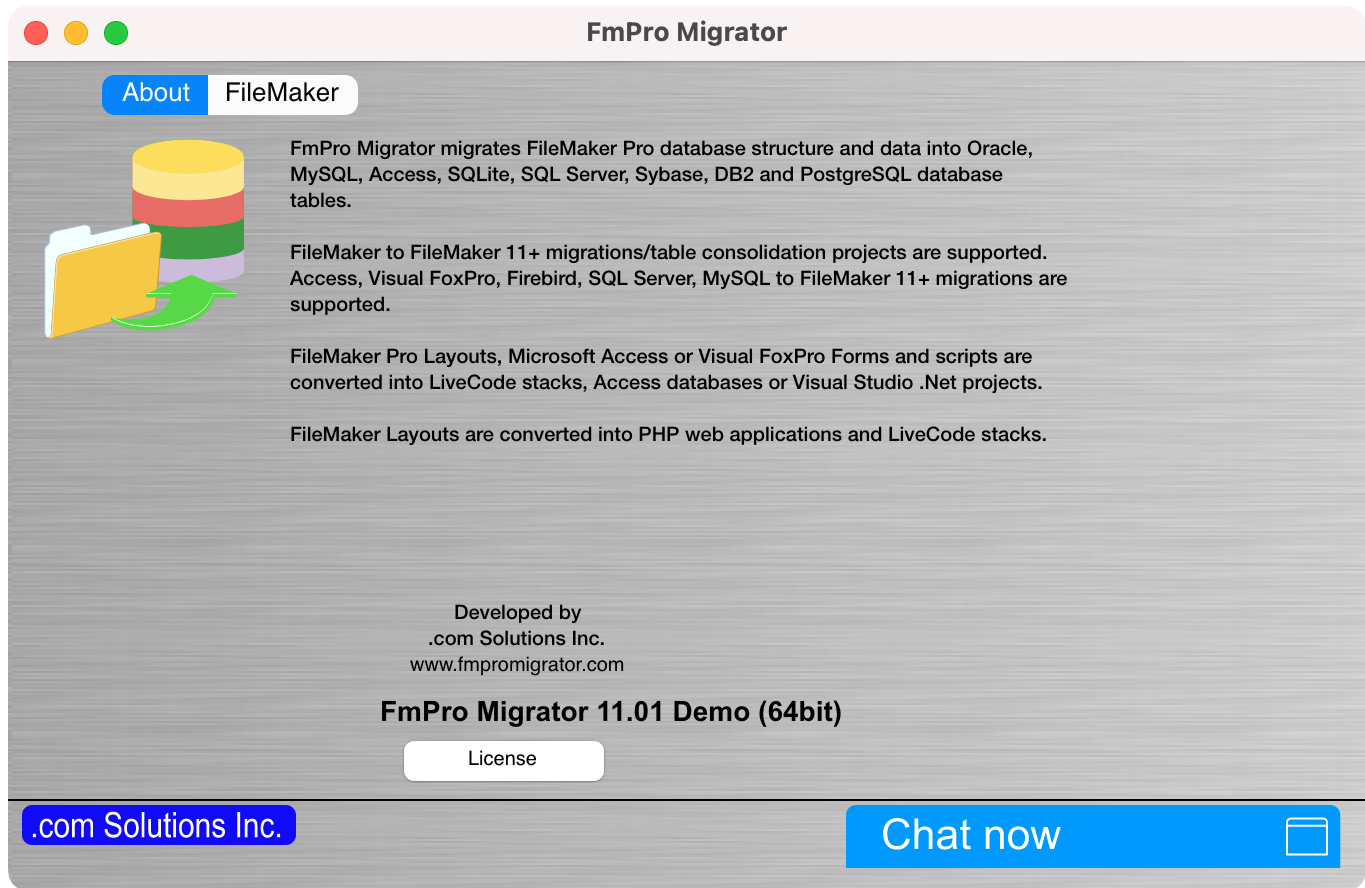
4/14/2024

### Demo Edition Dialog

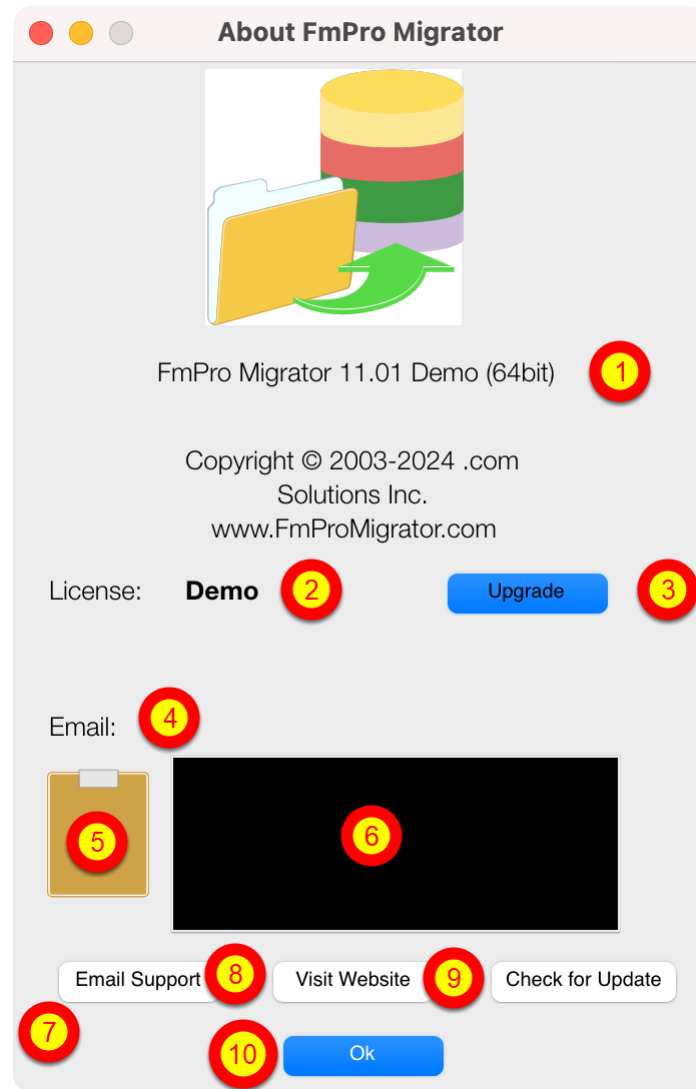


When launched the first time, FmPro Migrator will be running in Demo mode as shown in this screenshot. Clicking the Ok button opens the order page of the website.

In addition to transferring data for 5 database fields, the conversion features shown on the GUI tab of the Migration Process window will convert 5 layouts or forms/reports & scripts.



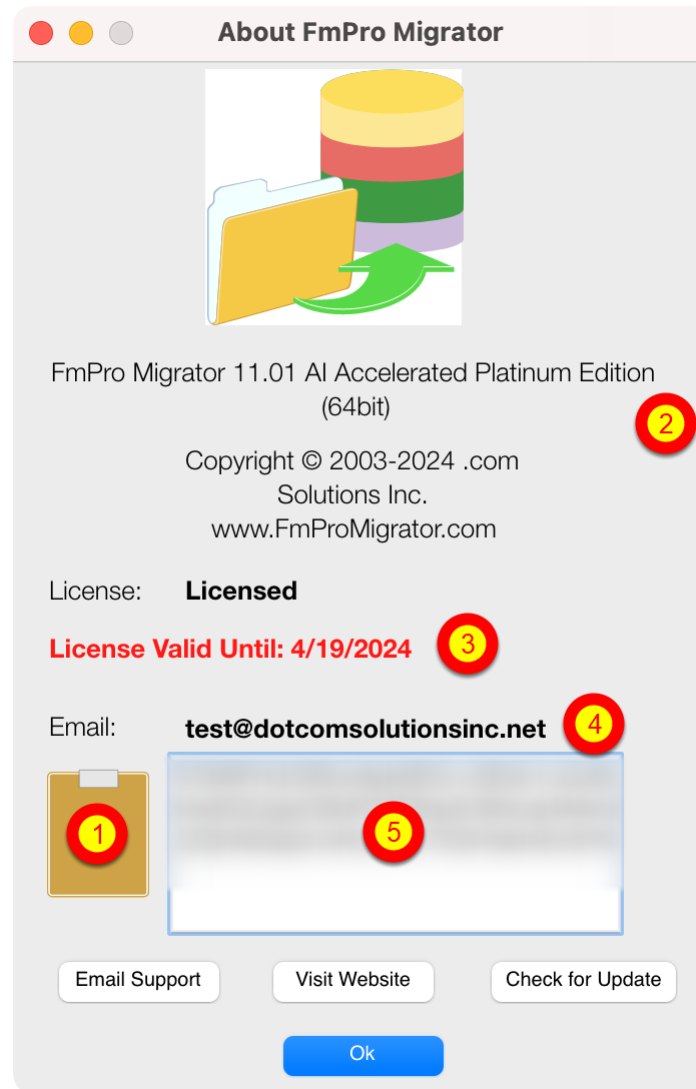
Clicking the "License" button on the About tab, or selecting the Help -> License/About menu items will open the About window where you can enter the license key.



On the About FmPro Migrator window:

(1) Product name and version licensed, (2) Demo or Licensed will be displayed, (3) The Upgrade button opens the website order page, (4) Email address associated with your license key, (5) Clipboard button reads the license key from the clipboard, (6) License key field, (7) Email Support button opens the Contact form on the website, (8) Visit Website opens the order page in Demo mode or the product page in Licensed mode, (9) Check for Update opens the downloads page with the latest software version, (10) Ok button closes the About window.

## Pasting License Key



Your product license key will be displayed in your web browser when your order has been completed. It will also will be sent via email at the same time, please check your SPAM folder if the email doesn't arrive in a few minutes.

(1) Copy the license key to the clipboard and click the clipboard icon. Once validated (2) the product name will change, (3) the License validation date will be updated, (4) your email address will be displayed, and (5) the license key will be displayed in the field under the Email address when clicking inside the field. In this screenshot the license key has been obscured.

And that is all you need to do to license all of the features of FmPro Migrator.

## Visual FoxPro Conversion - Step 1 - Get Info

---

This document explains the process of importing a Visual FoxPro project into FmPro Migrator. Once the metadata has been imported into FmPro Migrator, the Visual FoxPro project can then be converted into any of the database development environments supported by FmPro Migrator.

The Visual FoxPro DBF file data can also be converted into a wide range of databases including: Microsoft Access, DB2, FileMaker Pro, FrontBase, MySQL, OpenBase, Oracle, PostgreSQL, SQLite, SQL Server or Sybase.

As part of the automated conversion process, the PRG and Form code can be converted into over 50 different programming languages with the VFP Code Conversion Workbench - included with the AI Accelerated version of FmPro Migrator Platinum Edition.

Document Version 10

Updated for FmPro Migrator Platinum Edition 11.01

Added new Licensing FmPro Migrator info, updated screenshots

4/13/2024

### Pre-Migration Tasks

Before starting the Visual FoxPro Conversion process:

1) Add any Free Tables in the Visual FoxPro project to a new or existing database file.

The VFPEXport utility only exports table info from DBC files, in order to reduce the change of duplicate table names being overwritten during the dat transfer process. If the project contains duplicate table names within separate DBC files, one version of a duplicated table name should be renamed in order to insure uniqueness during the SQL database transfer process. All tables will be created within the same SQL database of a SQL database server, so duplicate tables would get dropped and overwritten.

2) If the Visual FoxPro solution contains only Free Tables, then this means that there won't be any relationships available for exporting. If you want relationships to be exported and used in the new solution, you will need to add all of the Free Tables into a DBC file and also create relationships between the tables within the DBC. This process will insure that the VFPEXport utility will be able to find and export the relationships between the tables. Even if the data is already located within a SQL database server it can be useful to create relationships in the DBC file for use by FmPro Migrator during the Forms conversion process.

3) In order to prepare the FoxPro 2.6 project for import into FmPro Migrator, perform the following steps:

A) Install Visual FoxPro 9.

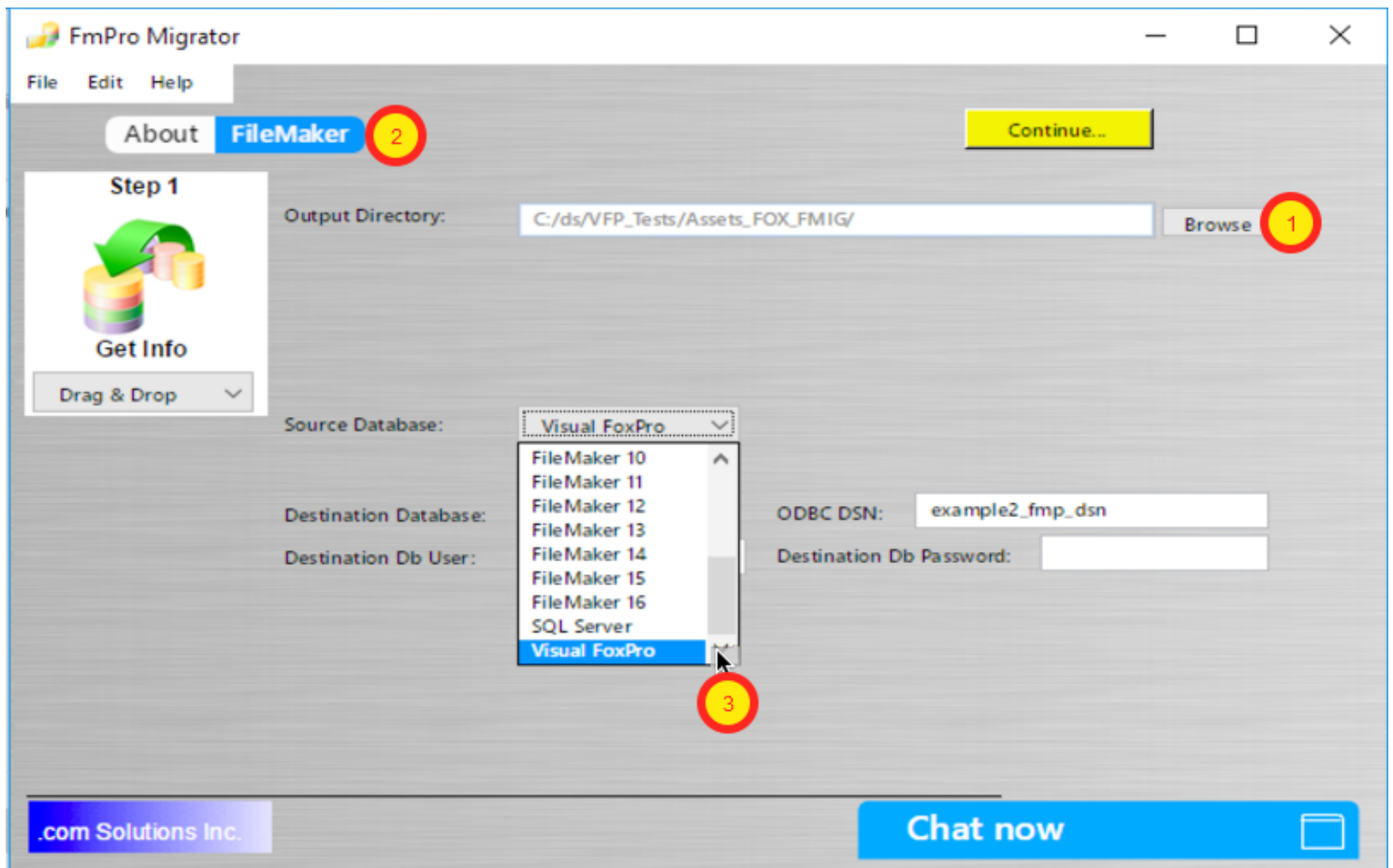
B) Make a copy of the FoxPro 2.6 project.



- C) Open and convert the copied project in Visual FoxPro 9. This will upgrade all of the files including the DBF files. This is why you want to work with a copy of the project - because the DBF files won't be readable in FoxPro after conversion. And you might need to continue developing the existing FoxPro app while performing your conversion.
- D) Once the FoxPro 2.6 project has been converted into Visual FoxPro 9, create a DBC and add all of the DBF files to the DBC.
- E) Manually add 1 empty form to the Visual FoxPro 9 project.

Now you are ready to follow the remaining info within this manual showing how to import the project into FmPro Migrator for conversion.

### Select Output Directory, Source Database Type



Launch FmPro Migrator, (1) click on the FileMaker tab, select the Output Director, (3) select Visual FoxPro from the Source Database menu.

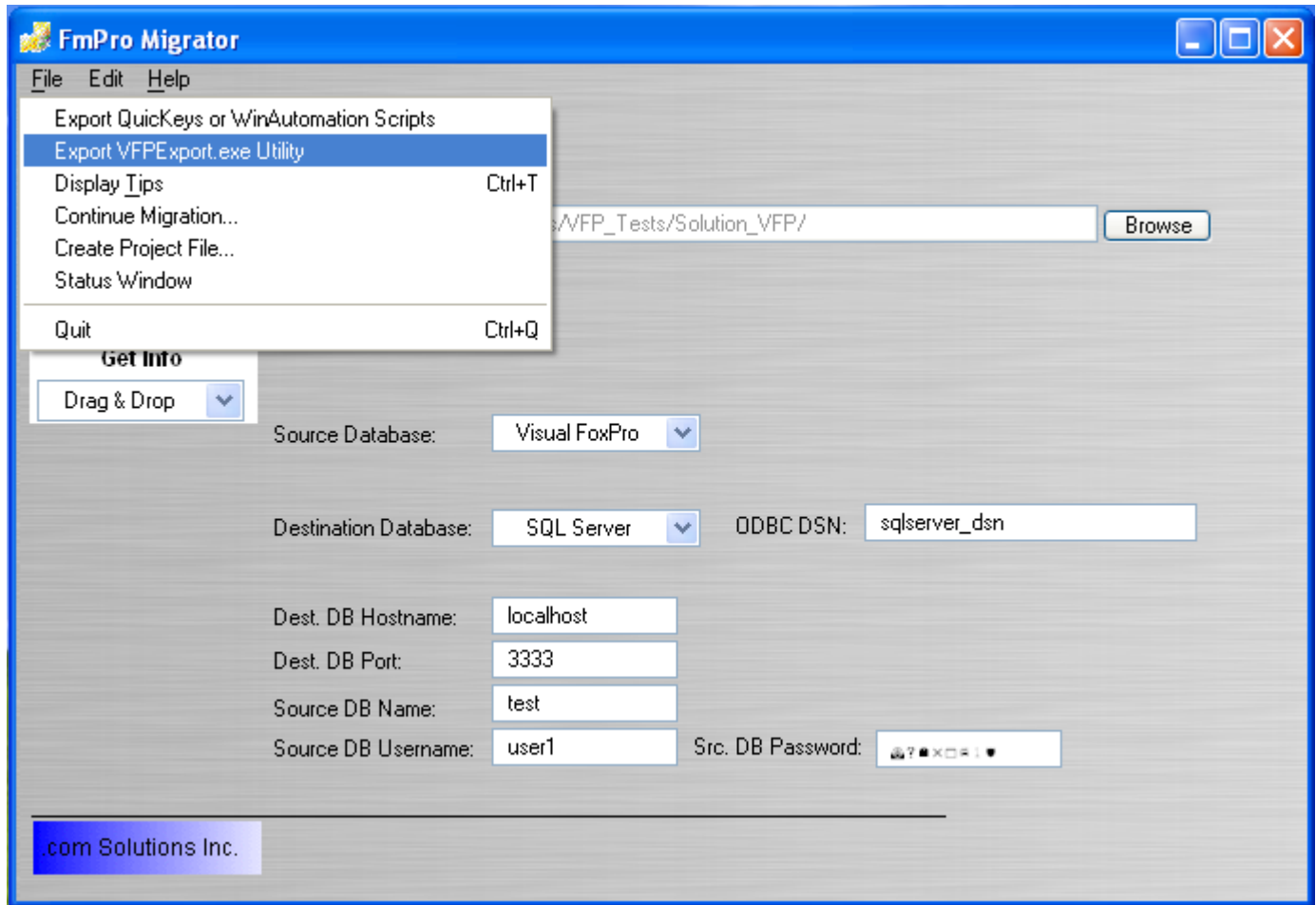
The Output Directory will be used by FmPro Migrator to store the FmPro Migrator MigrationProcess.db3 project file, as well as files generated during the conversion process. It is

not recommended that the Visual FoxPro project directory be used as the FmPro Migrator Output Directory. You will have the opportunity to select the Visual FoxPro project directory later in the conversion process.

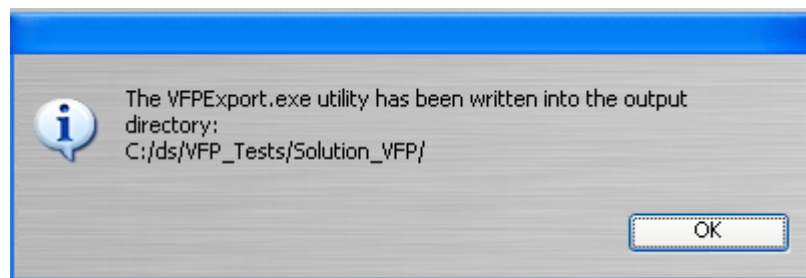
## Select Destination Database

(1) Select the destination database, enter the connection information for the destination database, (2) ODBC DSN, (3) hostname, (4) port, (5) database name, (6) login username/schema, (7) login password.

## Export the VFPEXport.exe Utility

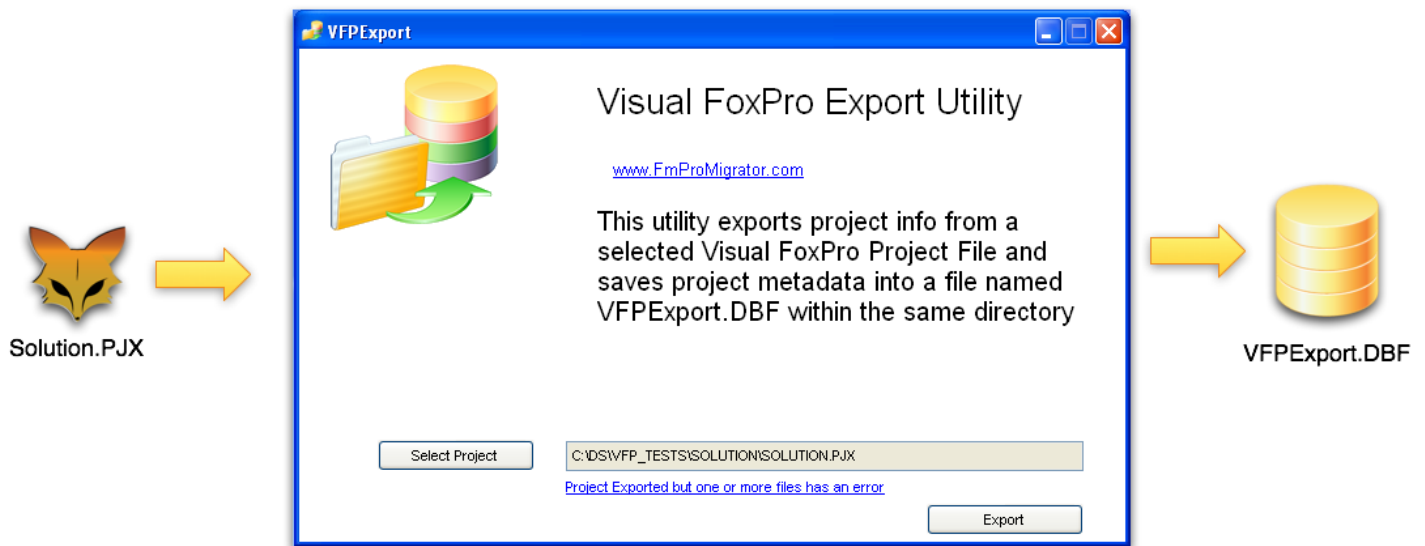


FmPro Migrator uses the included VFPEXport.exe utility to export metadata from Visual FoxPro .PJX project files. Select File -> Export VFPEXport.exe Utility from the File menu. The VFPEXport.exe utility is a compiled Visual FoxPro executable which reads and exports the metadata from a Visual FoxPro .PJX file. The resulting exported data is stored within the VFPEXport.DBF database file.



The VFPEXport utility will be saved into the FmPro Migrator Output Directory.

## VFPEXport Utility - Static MetaData Export Process



FmPro Migrator uses a 2 step process to import the metadata from within Visual FoxPro projects. Thru the use of the included VFPEXport.exe Utility, Visual FoxPro project metadata is read directly from the VCX, SCX, FRX, DBC, DBF, MNX and PRG files referenced within the .PJX project file. This information is written into a new database file named VFPEXport.DBF.

### Launch the VFPEXport.exe Utility

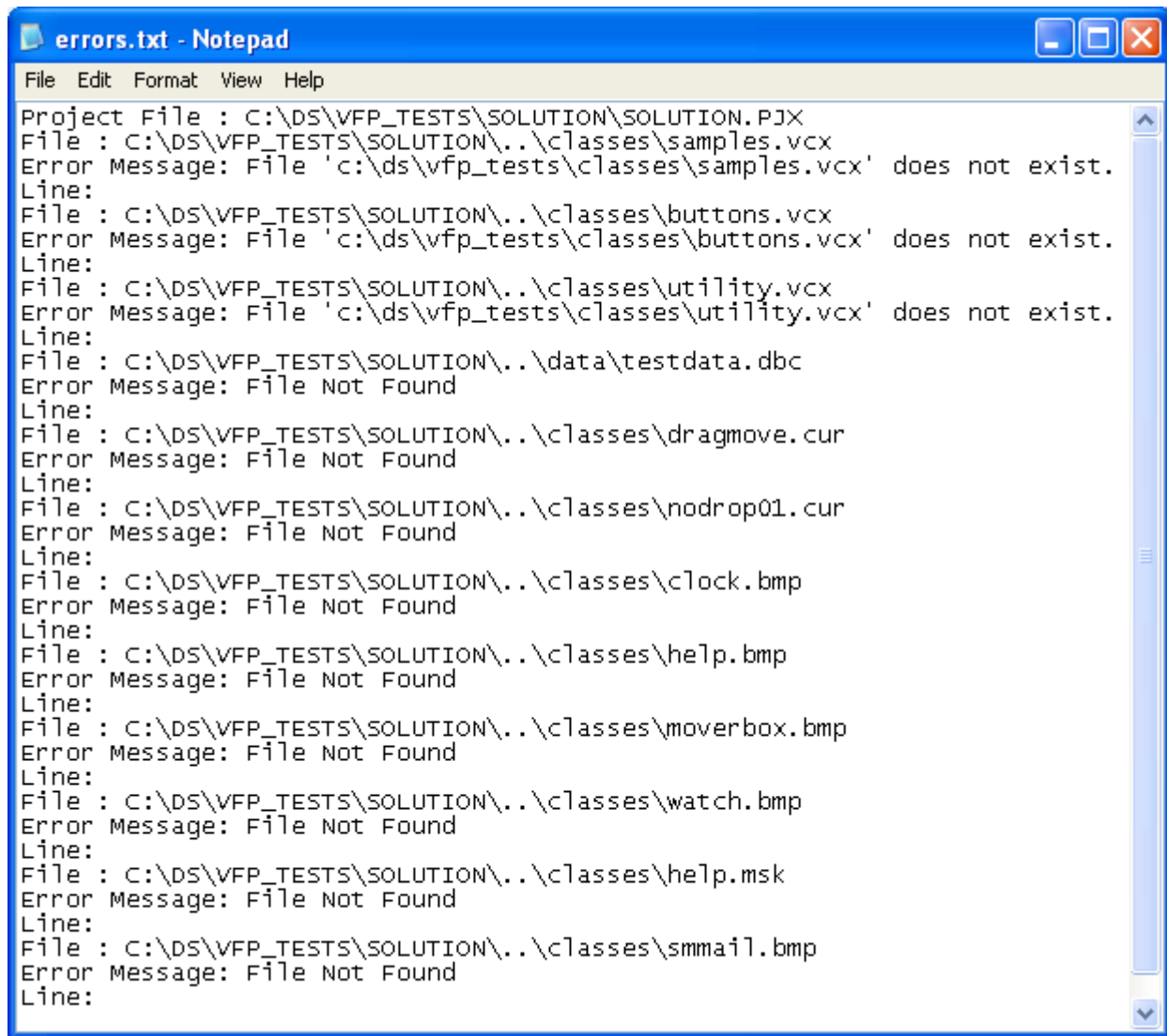


Launch the VFPEXport.exe utility, (1) Select the Visual FoxPro project file for the project to be converted, (2) click the Export button.  
(3) Any errors which were logged during the export process, will be listed in the errors.txt file which

will be written into the directory where the VFPEXport.exe utility is located.

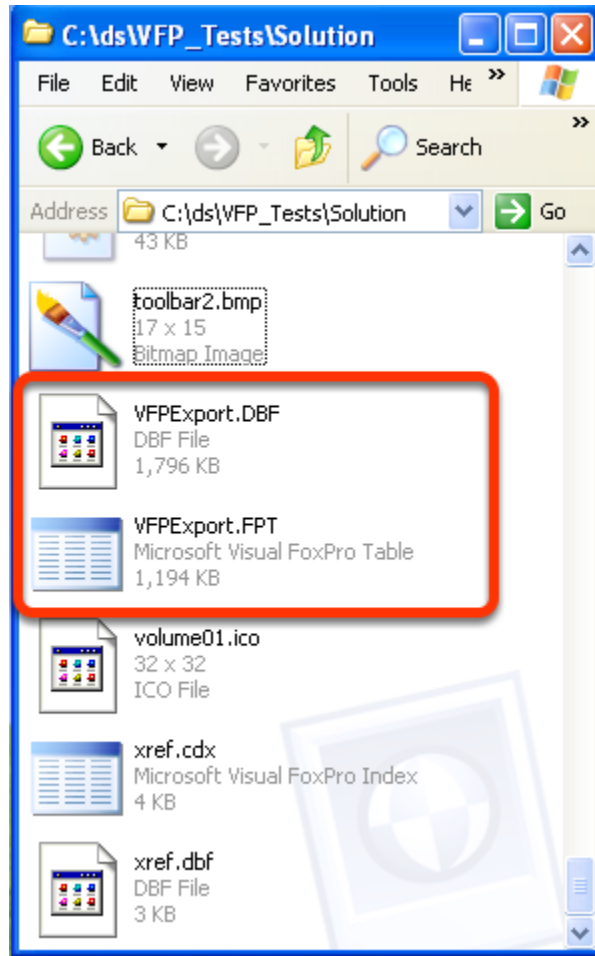
Close the VFPEXport.exe utility.

### VFPEXport Error Log File



```
errors.txt - Notepad
File Edit Format View Help
Project File : C:\DS\VFP_TESTS\SOLUTION\SOLUTION.PJX
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\samples.vcx
Error Message: File 'c:\ds\vfp_tests\classes\samples.vcx' does not exist.
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\buttons.vcx
Error Message: File 'c:\ds\vfp_tests\classes\buttons.vcx' does not exist.
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\utility.vcx
Error Message: File 'c:\ds\vfp_tests\classes\utility.vcx' does not exist.
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\data\testdata.dbc
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\dragmove.cur
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\nodrop01.cur
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\clock.bmp
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\help.bmp
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\moverbox.bmp
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\watch.bmp
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\help.msk
Error Message: File Not Found
Line:
File : C:\DS\VFP_TESTS\SOLUTION\..\classes\smmail.bmp
Error Message: File Not Found
Line:
```

Clicking the error log link will open the error log text file. A common source of errors is missing files which are listed in the .PJX file but which have been moved or deleted from the project directory.

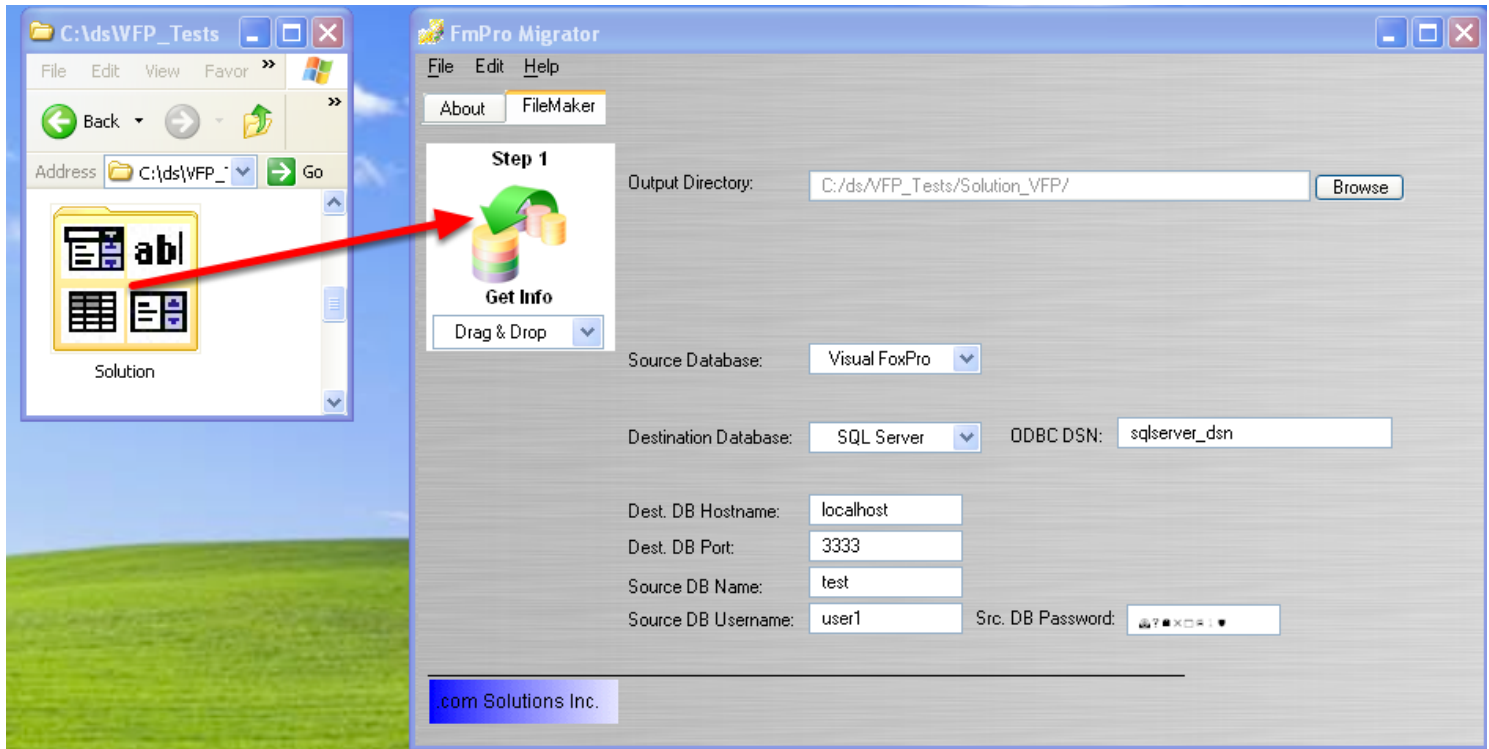


As a result of running the VFPEXport.exe utility, the VFPEXport.DBF/FPT files will be found within the Visual FoxPro project's directory. These files provide the Visual FoxPro project metadata which will be used by FmPro Migrator for the remaining steps of the conversion process.

### A Note for Experienced Developers

If you want to have more control over the final conversion process, you can edit the contents of the VFPEXport.DBF/FPT files prior to continuing the conversion process. All of the metadata for the Visual FoxPro project is contained within these two files. For instance, if you wanted to standardize the names of the fonts used in your project, you could edit the contents of the properties column of the DBF table to accomplish this task. Any changes to the VFPEXport.DBF/FPT contents must be done at this point in the processing, because the very next step of the processing will copy the contents of the VFPEXport.DBF/FPT files into a new VFPEXport.db3 SQLite database file. The SQLite version of the data will be used by FmPro Migrator for the conversion processing.

## Step 1 Get Info Button - Drag & Drop Visual FoxPro Project Directory



Drag & Drop the Visual FoxPro project directory onto the FmPro Migrator Step 1 Get Info button. FmPro Migrator will look for the VFPEXport.DBF file at the top-level of the Visual FoxPro project directory. When the VFPEXport.DBF file has been located, FmPro Migrator will copy the data from the VFPEXport.DBF/FPT files into a new SQLite database file named VFPEXport.db3.

During the conversion process, FmPro Migrator will open the console window 3 times as it is processing the files.

## Visual FoxPro Tables List

**Migration Process**

Tables: 7

Original Table	New Table	Source	Destination	Fields	Record
dvdprice	dvdprice	Visual FoxPro	SQL Server	3	
dvds	dvds	Visual FoxPro	SQL Server	7	
genres	genres	Visual FoxPro	SQL Server	2	
ids	ids	Visual FoxPro	SQL Server	2	
table001	table001	Visual FoxPro	SQL Server	16	

Fields: 3

ID	Original Fieldname	PK	Auto-Increment	Next SN	New Fieldname
1	id	0	0		
2	price	0	0		
3	listprice	0	0		

**Instructions:**

**Step 1:**  
Select Export VFPEXport.exe Utility from the File menu. The VFPEXport.exe utility will be written into the output directory.

**Step 2:**  
Launch the VFPEXport.exe utility, select the Visual FoxPro project (PJX) file, click the Export button. The metadata for the Visual FoxPro project will be exported into the VFPEXport.DBF file.

**Step 2: Create Table**  
Not Started

**Step 3: Transfer Data**  
Not Started

com Solutions Inc.

After the completion of the Step #1 processing, FmPro Migrator will automatically open the Migration Process window and display the list of Tables/Fields which have been extracted from the Visual FoxPro project.



## Visual FoxPro Relationships List

**Migration Process**

Instructions: Relationships: 6

**Relationships**

**Migration - FileMaker to SQL Databases:**

Within FileMaker Pro 7+ export a DDR report if using FileMaker Developer/Advanced. FileMaker Pro users may manually add Relationships and Predicates on this screen. Click the Import Relationships button to import a DDR xml file for each FileMaker database file, in order to obtain a complete list of all Relationships, Table Occurrences and File References. Once this info has been imported, delete duplicate Table Occurrences. Select all of the Relationships, then click the Create SQL

Left Table Name	Right Table Name	Predicate Count	LT:
customer	orders	1	0
orders	orditems	1	0
orditems	products	1	0
customer	orders	1	0
customer	orders	1	0

Left Table: customer

Right Table: orders

Cascade Create: ☐ Cascade Delete: ☐

Relationship SQL Code:

Predicates: 1

Left Table::Field	Join Type	Right Table::Field
customer::cust_id	=	orders::cust_id

com Solutions Inc.

The relationships exported from the Visual FoxPro project are listed under the Relationships tab of the Migration Process window. Once the tables have been created in the destination database, the SQL code can be created and run in order to create the relationships in the destination database.

## **Step 2 - Create Table**

## Step 2 - Create Table





If errors occur when creating the table, you can examine or change the table creation SQL code by clicking the Create Table Details button.

You can also change the destination database connection properties on the Step 1 tab in FmPro Migrator. Then click the Create Table button again.

Double-clicking the name of the table in the list of tables opens the Table Details window, where changes can be made directly. But it is easier to make database connection changes on the Step 1 tab since the changes will be made for all of the tables instead of just one individual table.

## Table Creation Details Window

**Table Creation Details**

This window displays the table creation SQL code which is executed in the destination database. Changes made in this window can be immediately executed in the destination database by clicking the Execute SQL buttons.

Drop Table Name:

Table Creation SQL Code:

```
CREATE TABLE [Assets].[user1].[dvdprice]
(
  id
  VARCHAR(25)                NOT NULL,
  price
  DECIMAL(18,4)              NOT NULL,
  listprice
  DECIMAL(18,4)              NOT NULL
)
```

Drop Table

Execute SQL

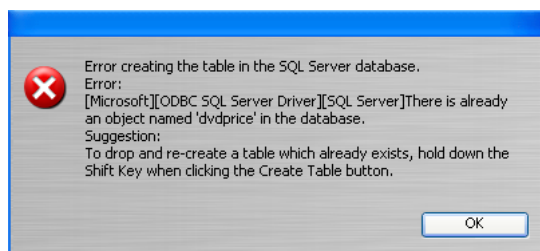
Save

Post Table Creation SQL Code:

Execute SQL

Save

## Duplicate Table Warning Dialog



If the table already exists in the destination database, the duplicate table error dialog will be displayed. Hold down the shift key when clicking on the Create Table button in order to drop and recreate the table.

**Note:** Dropping the table will cause the loss of all data in the table.

# Step 3 - Transfer Data

## Visual FoxPro Conversion - Step 3 - Transfer Data

### Step 3 - Transfer Data

The screenshot shows the 'Migration Process' window with the 'Tables' tab selected. The 'Instructions' pane on the left provides guidance for Step 1 and Step 2. The main area displays a table of migration details and a progress indicator at the bottom.

**Instructions:**

**Step 1:**  
Select Export VFPEXport.exe Utility from the File menu. The VFPEXport.exe utility will be written into the output directory.

**Step 2:**  
Launch the VFPEXport.exe utility, select the Visual FoxPro project (PJX) file, click the Export button. The metadata for the Visual FoxPro project will be exported into the VFPEXport.DBF file.

**Tables:** 7

Original Table	New Table	Source	Destination	Fields	Record
dvdprice	dvdprice	Visual FoxPro	SQL Server	3	
dvds	dvds	Visual FoxPro	SQL Server	7	
genres	genres	Visual FoxPro	SQL Server	2	
ids	ids	Visual FoxPro	SQL Server	2	
table001	table001	Visual FoxPro	SQL Server	16	

**Fields:** 3

ID	Original Fieldname	PK	Auto-Increment	Next SN	New Fieldname
1	id	0	0		id
2	price	0	0		price
3	listprice	0	0		listprice

**Step 2:** Create Table Completed

**Step 3:** Transfer Data Not Started

com Solutions Inc.

Click the Step 3 button to transfer data from a DBF file into the new table which has been created in the destination database.



## Troubleshooting - Column Change Buttons

The screenshot shows the 'Migration Process' window with the 'Tables' tab selected. It displays a table mapping table and a field mapping table. A red box highlights the 'Column Change' buttons (123, abc, Varchar, Text) for the 'id' field.

**Tables:** 7

Original Table	New Table	Source	Destination	Fields	Record
dvdprice	dvdprice	Visual FoxPro	SQL Server	3	
dvds	dvds	Visual FoxPro	SQL Server	7	
genres	genres	Visual FoxPro	SQL Server	2	
ids	ids	Visual FoxPro	SQL Server	2	
table001	table001	Visual FoxPro	SQL Server	16	

**Fields:** 3

ID	Original Fieldname	PK	Auto-Increment	Next SN	New Fieldname
1	id	0	0		id
2	price	0	0		price
3	listprice	0	0		listprice

**Column Change Options:** 123, abc, Varchar, Text

**Step 2:** Create Table Completed

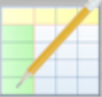
**Step 3:** Transfer Data Not Started


com Solutions Inc.

Sometimes, it is necessary to change the column types for table columns in order to transfer the data into the destination database. For instance, due to data entry errors or unvalidated data - a Date or Numeric column may need to be converted into a text column. This would be done just for the purpose of transferring the data as text into the destination database. Then stored procedures or SQL code could be written to fix problems with the entered data.

**Table Details**

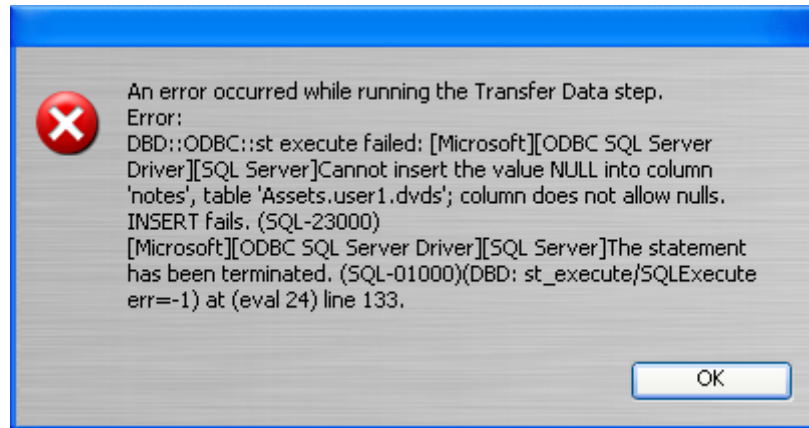
Change the table details using fields listed below. Changes occur when clicking outside the field.



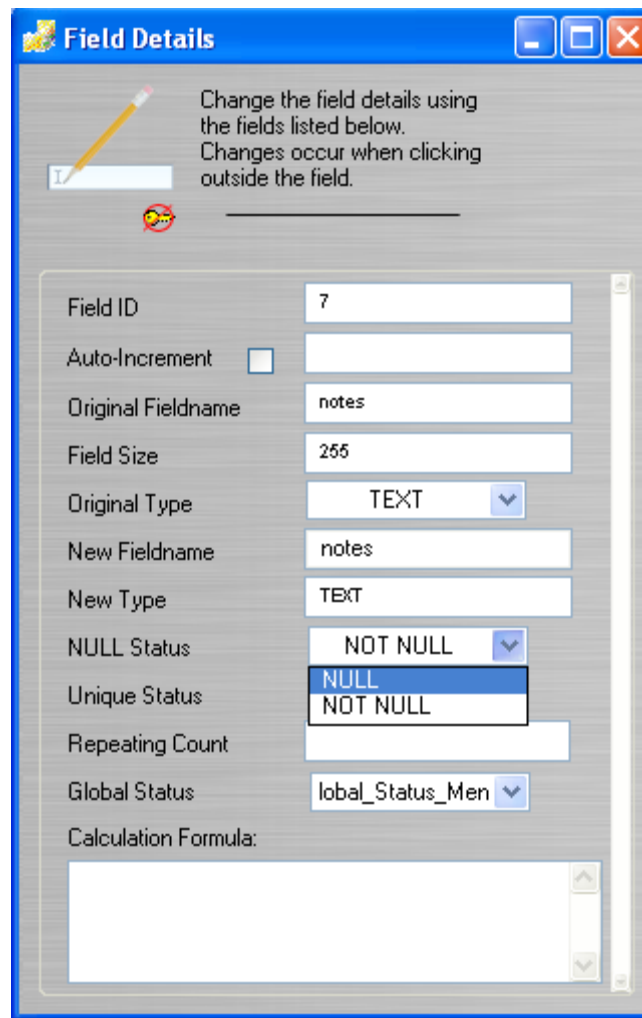


Original Table	dvdprice
New Table	dvdprice
Source DB	Visual FoxPro
Destination DB	SQL Server
Fields	3
Records	
Source DSN	
Source DB Username	
Source DB Password	
Destination DSN	sqlserver_dsn
Dest. DSN Username	
Dest. DSN Password	
Dest. DB Hostname	localhost
Dest. DB Port#	
Dest. DB Name	Assets
Dest. DB Username	user1
Dest. DB Password	user1pwd
Oracle Service Name	
Oracle User Name <input type="checkbox"/>	

The Table Details window can be displayed by double-clicking on a table in the list of tables. Clicking a different table with the Table Details window displayed will automatically update the info in the Table Details window with the info for the table which has been clicked.



Sometimes it is necessary to change column properties like the NULL/NOT NULL status. The Visual FoxPro dvds.dbf file within the example Solution includes columns set for NOT NULL, but the tables actually contain NULL values - thus causing an error during the data transfer step. Column properties can be changed via the Field Details window.



The image shows a Windows-style dialog box titled "Field Details". At the top left is a pencil icon and a text box containing the letter "I". To the right of this is instructional text: "Change the field details using the fields listed below. Changes occur when clicking outside the field." Below the text is a horizontal line. The main area of the dialog contains a list of field properties, each with a label and a corresponding input field or dropdown menu. The properties are: Field ID (text box with "7"), Auto-Increment (checkbox, unchecked), Original Fieldname (text box with "notes"), Field Size (text box with "255"), Original Type (dropdown menu showing "TEXT"), New Fieldname (text box with "notes"), New Type (text box with "TEXT"), NULL Status (dropdown menu showing "NOT NULL"), Unique Status (dropdown menu with "NULL" selected and "NOT NULL" as an option), Repeating Count (text box), Global Status (dropdown menu showing "lobal\_Status\_Men"), and Calculation Formula (a large empty text box). The dialog has standard Windows window controls (minimize, maximize, close) in the top right corner.

Field ID	7
Auto-Increment	<input type="checkbox"/>
Original Fieldname	notes
Field Size	255
Original Type	TEXT
New Fieldname	notes
New Type	TEXT
NULL Status	NOT NULL
Unique Status	NULL
Repeating Count	
Global Status	lobal_Status_Men
Calculation Formula:	

Double-clicking on a column for the database table opens the Field Details window.

After making changes to column properties, hold down the shift key then click the Create Table button in order to drop and recreate the table. Then click the Transfer Data button again.

## Report Export Buttons

The screenshot shows the 'Migration Process' window with the 'Tables' tab selected. The 'XLS' button in the top toolbar is highlighted with a red rectangle. The window displays a table of migration data and instructions on the left.

**Instructions:**

**Step 1:**  
Select Export VFPEXport.exe Utility from the File menu. The VFPEXport.exe utility will be written into the output directory.

**Step 2:**  
Launch the VFPEXport.exe utility, select the Visual FoxPro project (PJX) file, click the Export button. The metadata for the Visual FoxPro project will be exported into the VFPEXport.DBF file.

Original Table	New Table	Source	Destination	Fields	Record
dvdprice	dvdprice	Visual FoxPro	SQL Server	3	
dvds	dvds	Visual FoxPro	SQL Server	7	
genres	genres	Visual FoxPro	SQL Server	2	
ids	ids	Visual FoxPro	SQL Server	2	
table001	table001	Visual FoxPro	SQL Server	16	

Fields: 7

ID	Original Fieldname	PK	Auto-Increment	Next SN	New Fieldname
1	id	0	0		id
2	name	0	0		name
3	listprice	0	0		listprice
4	duedate	0	0		duedate
5	genreid	0	0		genreid

**Step 2**  
Create Table  
Completed

**Step 3**  
Transfer Data  
Completed

© 2008 Solutions Inc.

The XLS report button exports a summary .XLS spreadsheet to the output directory containing the table names and completion status for each of the tables in the list.

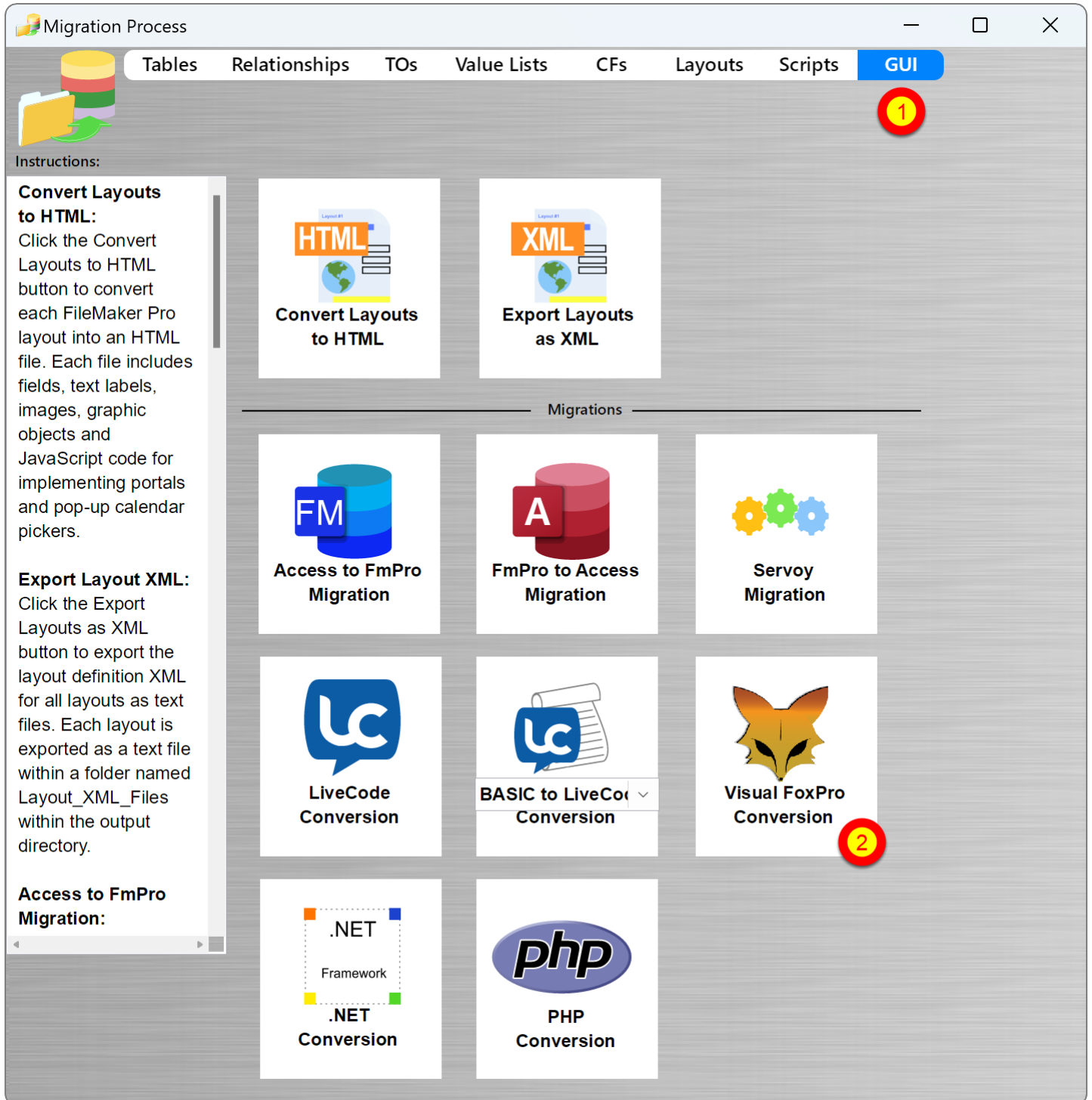
**Note:** The completion status below the Step 2/Step 3 buttons is a pop-up menu which can be changed manually. Therefore if you want to mark a status as N/A then this can easily be done for any of the DBF tables.

The 2nd report button, creates a detailed table summary report for the currently selected table. This plain text file includes info about the column types/sizes and is also written into the output directory.

# Import MetaData

## Visual FoxPro Conversion - Import MetaData

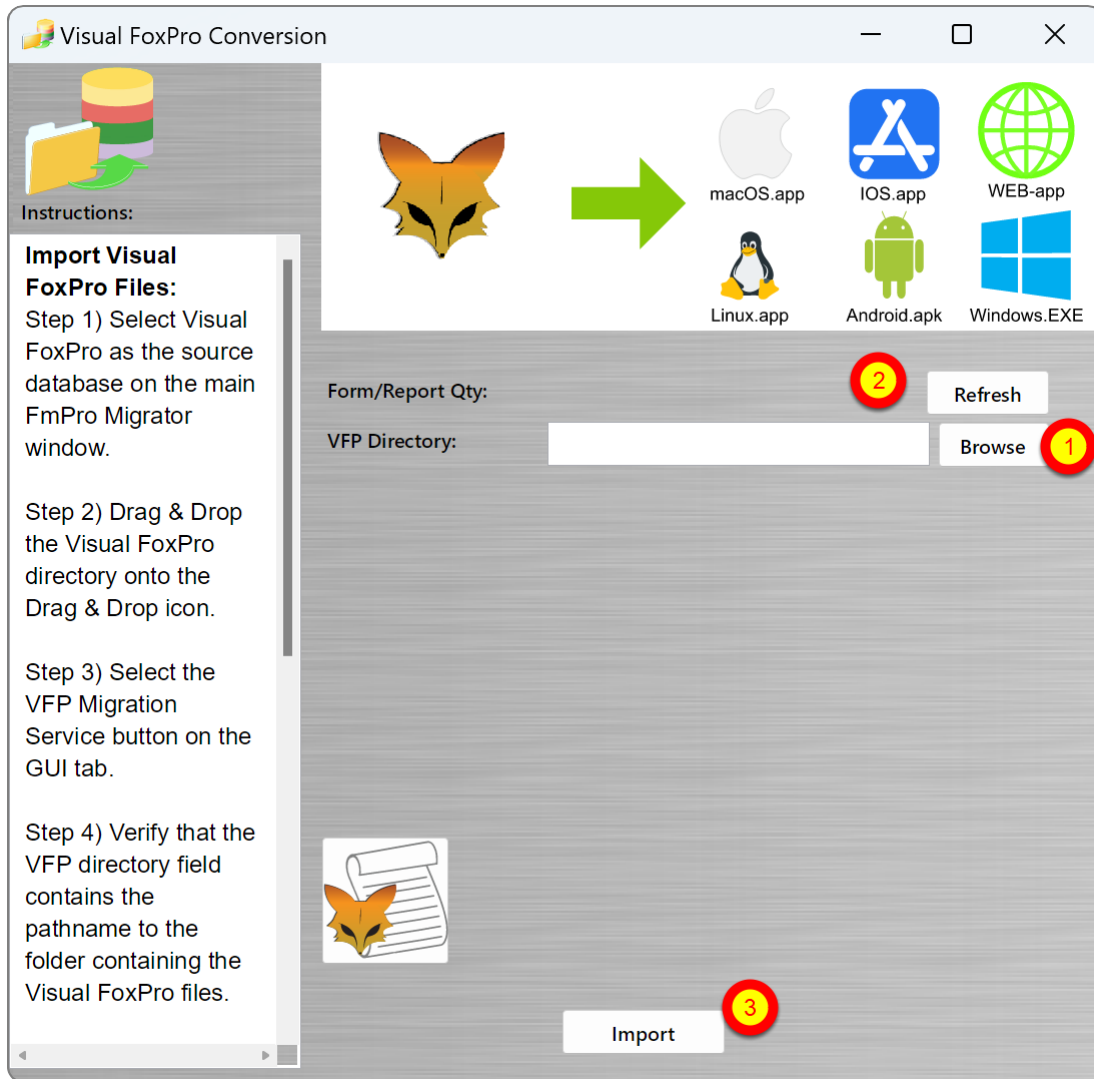
### GUI Tab - Visual FoxPro Conversion Button



(1) Click the GUI tab on the Migration Process window, (2) click the Visual FoxPro Conversion button.



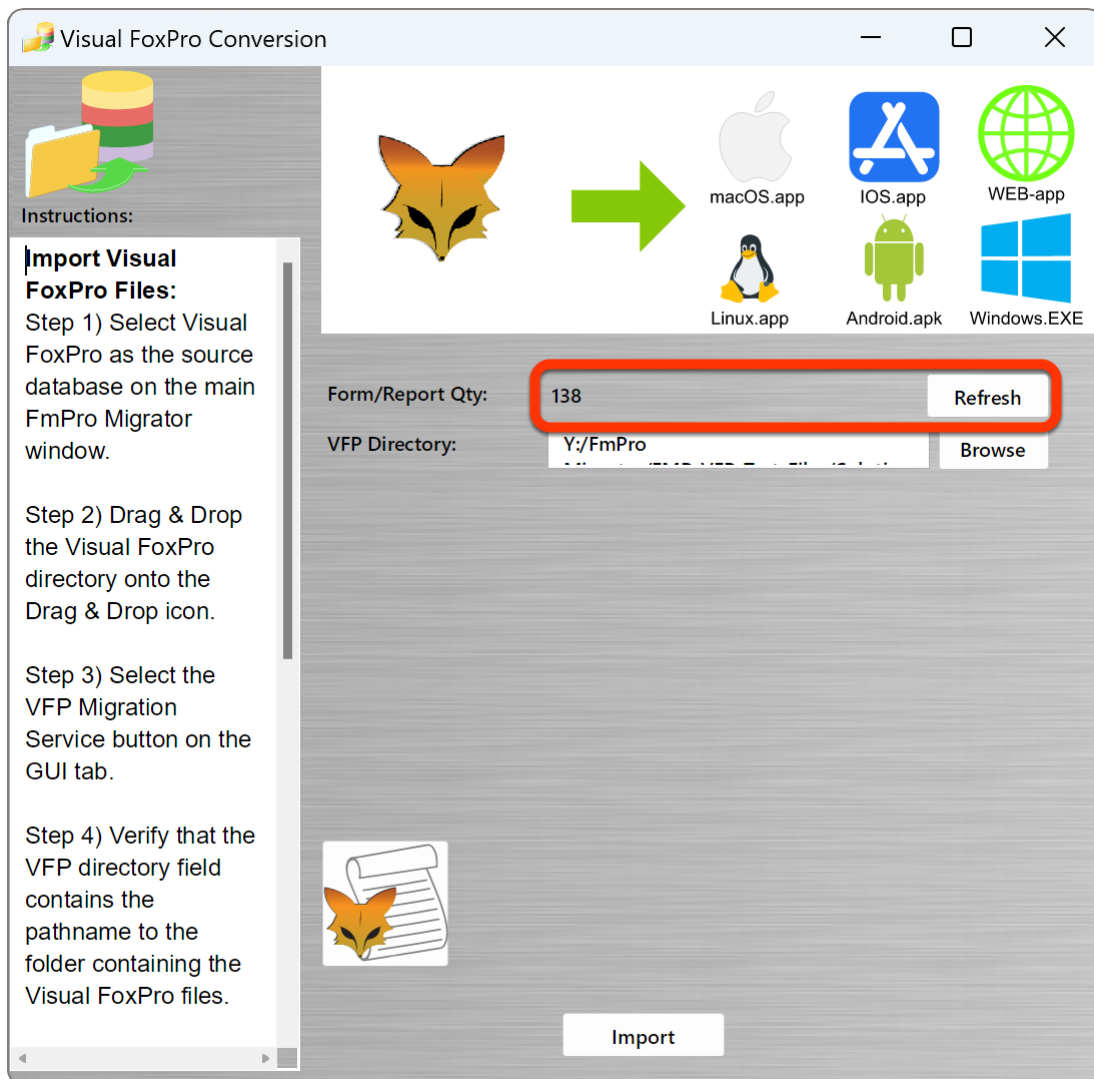
## Visual FoxPro Conversion Window - Features



(1) Select the Visual FoxPro project directory, (2) click the Refresh button to update the list of Forms/Reports in the Visual FoxPro project. (3) Click the Import button to convert and import and the Visual FoxPro project metadata into FmPro Migrator.

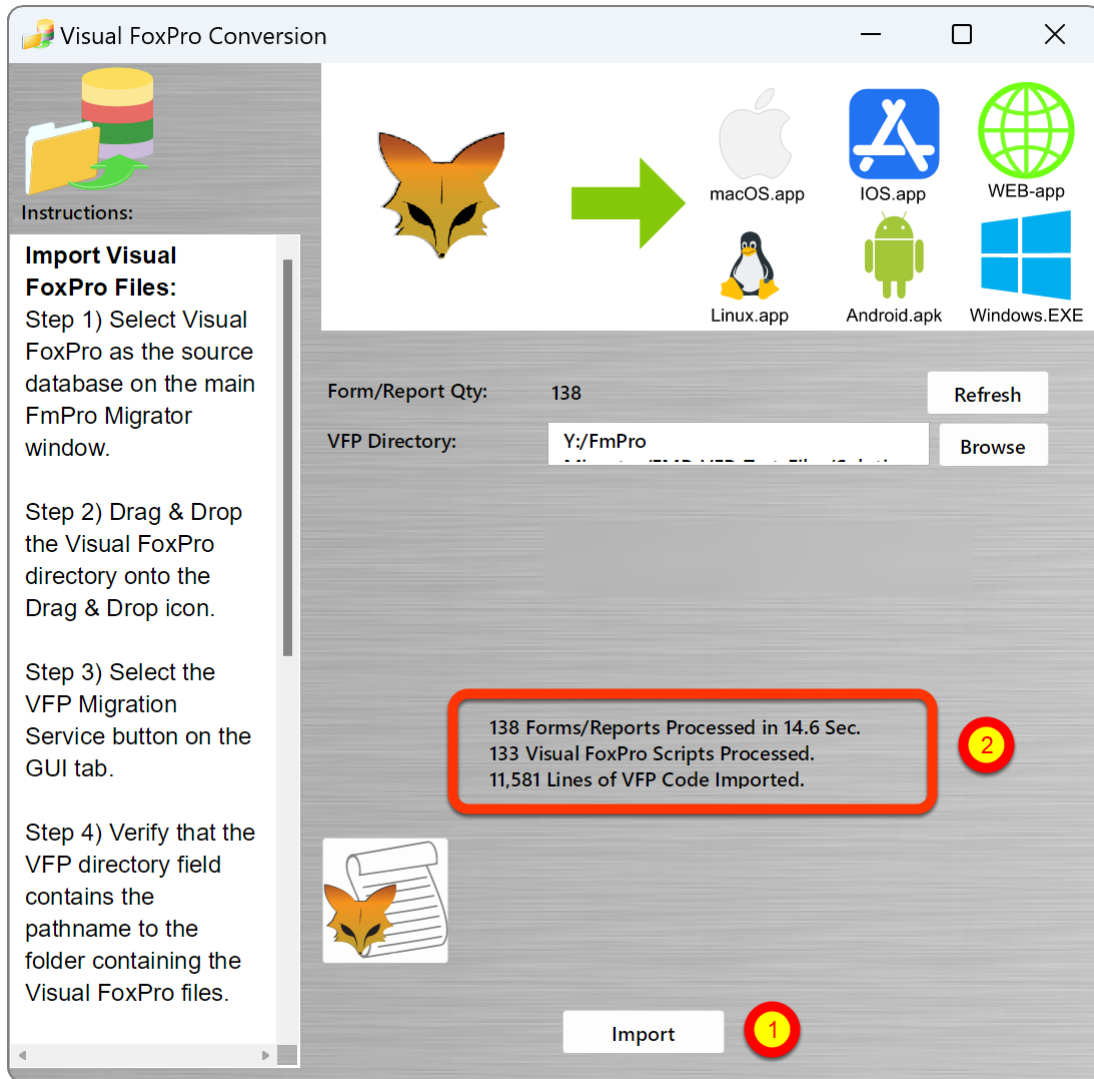
**Note:** In Demo mode, up to 5 Forms/Reports will be imported.

## Visual FoxPro Conversion Window - Refresh Button



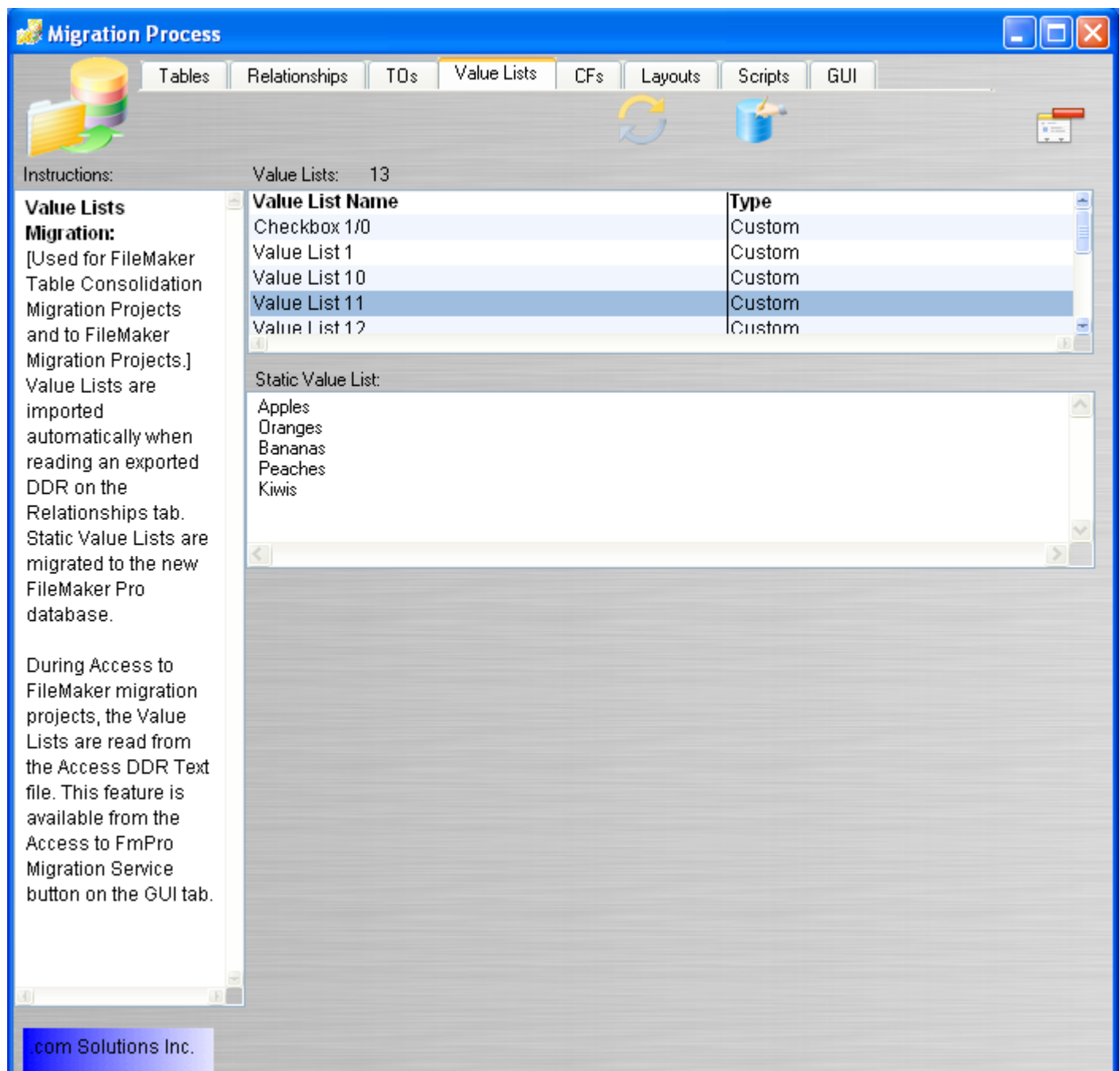
After selecting the Visual FoxPro project directory, the number of Forms/Reports is read from the VFPEXport.db3 SQLite database and displayed on the Visual FoxPro Conversion window. [It is generally no necessary to click the Refresh button anymore.]

## Visual FoxPro Conversion Window - Import Button



(1) Clicking the Import button will start the processing of the Forms/Reports and their enclosed objects. FmPro Migrator converts each Form/Report and enclosed objects into the Layout XML format which is used to store the information within FmPro Migrator. Any comboBox static values which are found on the Forms will be converted into Value Lists. And any FoxPro code embedded within Form/Report objects will also be imported into FmPro Migrator as well as being written into the VFP\_Scripts.txt file within the FmPro Migrator Output Directory.

## Visual FoxPro Conversion Results - Value Lists



Static lists of values from comboBox objects will be converted into Value Lists. Clicking on a Value List displays its contents as a list of text values.

## Visual FoxPro Conversion Results - Layouts & Portals

**Migration Process**

Instructions: Layouts: 87 dyngrid Table: products Pass#1

**Layout Migration:**  
[Used for Table Consolidation, Access to FileMaker Pro and FileMaker Layout to HTML, Servoy, and Access migration projects.] Within FileMaker 7/8/9/10 (layout mode), select all objects on a layout then select Copy from the Edit menu. Click the Add Layout button on this screen to copy the layout definition from the clipboard into FmPro Migrator. Each layout can be copied individually in this manner, or all layouts can be imported in batch mode (see info below). Each layout is given a default name in the form: Layoutxx where xx is an incrementing

Layout Name	Size	Fields	Tables
ddcwlrx	606,390	0	0
ddrop	490,320	0	0
dsession	749,273	4	0
dyngrid	558,498	13	1
editbox	519,498	4	0
fillmenu	536,382	3	0

Portals: Portal1

Portal Name	Fill Method	Field Titles
Portal1	On Record Advance	

Layout Image

.com Solutions Inc.

Forms and Reports are converted into Layouts, visible under the Layouts tab. The selected Layout shown in this screenshot shows a portal, which was created from the Visual FoxPro Grid object. Clicking on the portal shows its size and position within the rectangle of the Layout Image. The exact contents of the Layout will be visible once the Layouts have been converted into the destination database/development environment.

## Visual FoxPro Conversion Results - Scripts

**Migration Process**

Instructions: Scripts: 131

**ScriptMaker Script Migration:**  
Copy scripts from FileMaker Pro 9 or FileMaker Advanced onto the clipboard, from within FileMaker. Click the Add Scripts button on this screen to add the scripts to FmPro Migrator. All scripts within a FileMaker file can be copied all at once, each script will retain its name as displayed within ScriptMaker.

During Access to FileMaker Pro migration projects, the Visual Basic code is read from the Access DDR Text file, converted into FileMaker Scripts and displayed on this tab. This feature is available from the Access to FmPro

Script Name	Count	Script Text
Form_databook	1307	
Form_dbctofix	671	
Form_ddrop	478	
Form_dsession	805	
Form_dyngrid	1513	
Form_editbox	772	

**Script Text:**

```
# ----- Form: dsession
# ----- File: C:\DS\WFP_TESTS\SOLUTION\ffc\dsession.scx
# ----- Object: frmfd
# PROCEDURE drawaction
# Parameters nMPointer
# #define INACTIVE_LOC " - (Inactive - "
# #define ACTIVE_LOC " - (Active - "
# * Change mouse pointer and form caption as needed
#
# if nMPointer = 1
#   THISFORM.mousepointer = 2
# else
#   THISFORM.MousePointer = 1
# endif
#
# THISFORM.cFormMode = iif(nMPointer = 2, ;
#   INACTIVE_LOC,ACTIVE_LOC)
#
# THISFORM.SetCaption
# THISFORM.refresh
#
#
# ENDPROC
```

.com Solutions Inc.

Clicking the Scripts tab of the Migration Process window displays the source code which has been exported from the Visual FoxPro project. Visual FoxPro code for each Form/Report object is exported and grouped together with code from all of the other objects on the same Form/Report. Clicking a script displays its contents.

Source code from PRG files is also displayed in the list of scripts, with one script representing each .PRG file.

## Conversion Process - Next Steps

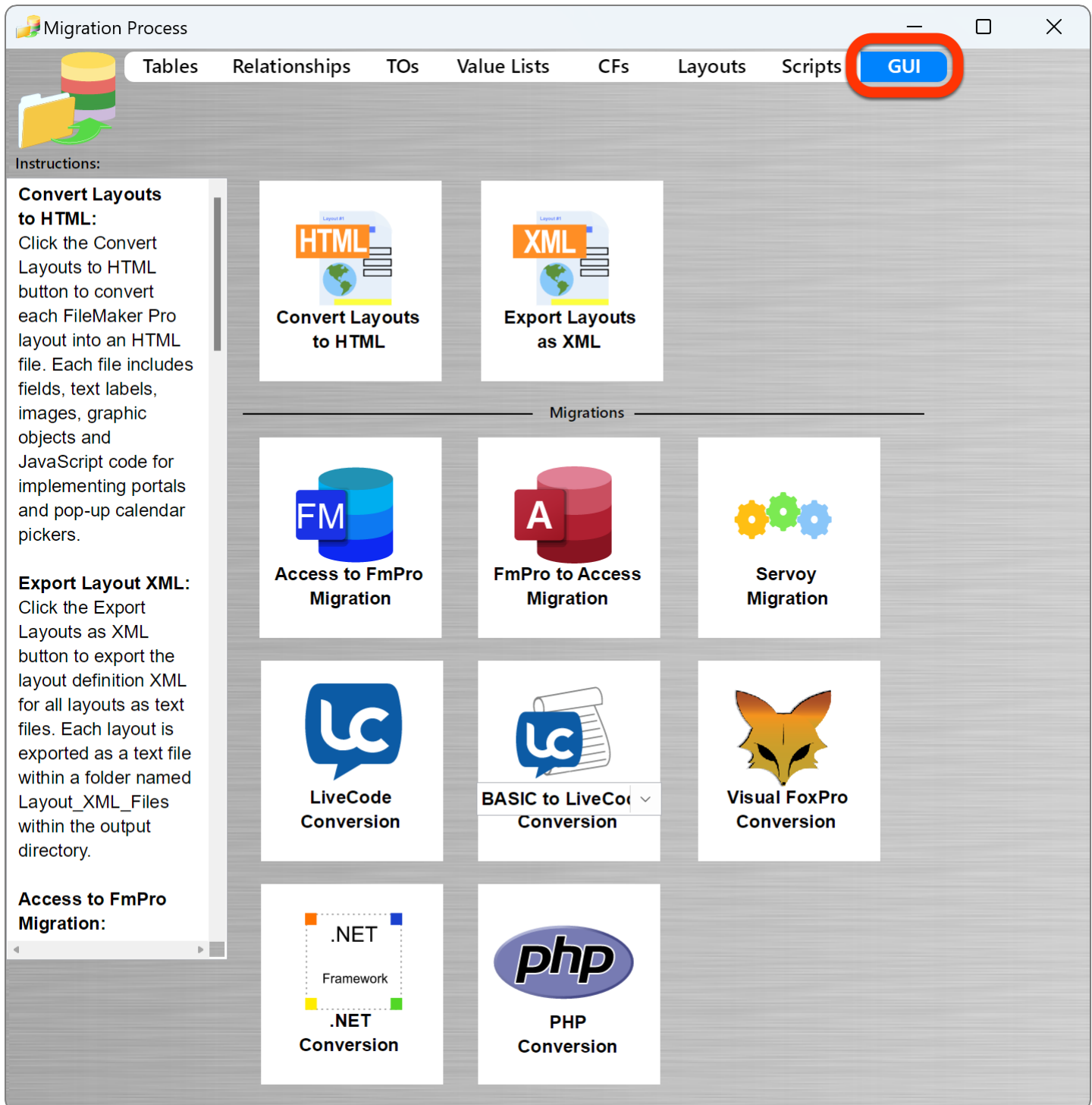
At this point in the conversion process, the metadata from the Visual FoxPro project has been imported into FmPro Migrator and stored within the FmPro Migrator MigrationProcess.db3 database file using a standardized format. This internal storage format makes it possible to perform conversions into any of the formats supported by FmPro Migrator. It is a matter of selecting the desired conversion target database/development environment and generating the files.

# Form/Report Conversion

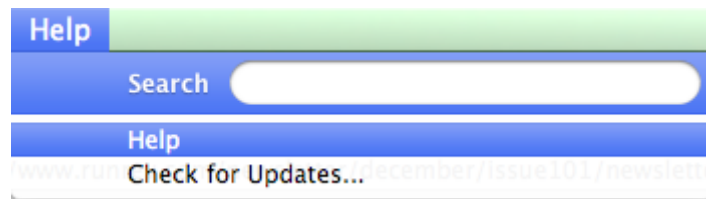


## Converting Forms/Reports and Layouts - GUI Tab

FmPro Migrator contains features on the GUI tab of the Migration Process window capable of performing conversions to a wide range of development environments. Click on any of the icons on the GUI tab to explore these options.



## Help Menu - Link to WebSite PDF Files



Additional info is available for each of these conversion options in separate YouTube video playlists and PDF files on the support web page. Select the Help menu item to visit the FmPro Migrator support web page.

# **Code Conversion with the VFP Code Conversion Workbench**

## Using the Code Conversion Workbench

---

The Code Conversion Workbench is included with the AI Accelerated version of FmPro Migrator Platinum Edition.

The Code Conversion Workbench consists of a single application window used to manage the conversion of hundreds or thousands of scripts into over 50 programming languages. As scripts are completed, they can be checked off in the grid of script on the left side of the window.

The source of scrips used by the Code Conversion Workbench is the MigrationProcess.db3 SQLite database file which is created and used by FmPro Migrator. Before performing code conversion, the scripts need to already be imported into the Scripts tab of the Migration Process window of FmPro Migrator.

Each source database is used to prefix the name of the code conversion, so if Visual FoxPro is the source database type, the title across the window will be: VFP Code Conversion Workbench.

Most screenshots in this manual are from macOS, but the tool works exactly the same way on Windows.

## Code Conversion Workbench Window

Code Conversion Workbench

# VFP Code Conversion Workbench

1 2

Completed: 2 of 6 Scripts Load Data

Status	ID	Size	Script Name
<input type="checkbox"/> Not Started	1	2441	pgraph.prg
<input type="checkbox"/> Not Started	2	4760	cgraph.prg
<input checked="" type="checkbox"/> Completed	3	6788	main.prg
<input type="checkbox"/> Not Started	4	14406	fdproc.prg
<input checked="" type="checkbox"/> Not Started	5	20234	datepicker.prg
<input checked="" type="checkbox"/> Completed	6	1207	frmanimation

3

Source Script (Editable): 14 Size: 2,059

```
PROCEDURE cmdok.Click
#DEFINE FROM_GREATER_TO_LOC "The from date must be less than or
equal to the to date."
#DEFINE FROM_MONTH_LOC "You must select a month to start."
#DEFINE FROM_YEAR_LOC "You must select or enter a year to start."
#DEFINE TO_MONTH_LOC "You must select a month to end."
#DEFINE TO_YEAR_LOC "You must select or enter a year to end."

PUBLIC dStart_Date,dEnd_Date

*! First, get the values the user entered into the combo boxes.
nFromMonth = VAL(THISFORM.cboFromMonth.value)
```

Command (Editable): 15

Translate this Visual FoxPro code into C# code

Converted Script: 17

Source: 4 Visual FoxPro

Output Language: 5 C#

Procedure/Function: 6 cmdok.Click 685-749

Section: 7 1

Convert 16

Vendor: 8 OpenAI

API Key: 9 dcsi

Model: 10 gpt-3.5-turbo

Tokens Used Today: 12 713

Output Filename: 13 datepick\_cmdok\_Click.cs

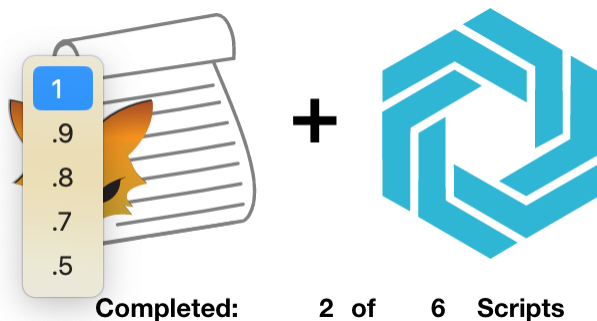
11 Refresh

18

Need Help?

Most features in this window include tooltips, described in this manual.

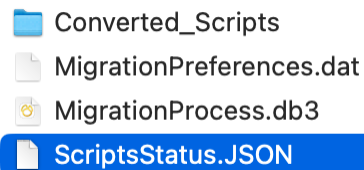
### 1) Window Scale Factor - Pop-up Menu



The Code Conversion Workbench window will auto-scale in size based upon the size of your display. You can change the window scaling by selecting a different value from this popup menu. The Window Scale Factor Menu tooltip will be displayed when you move the cursor over this

normally hidden popup menu.

## 2) Load Data Button



If there are scripts located within the FmPro Migrator MigrationProcess.db3 SQLite project file, the grid of scripts should be displayed automatically as soon as the window opens. If there weren't any scripts in the MigrationProcess.db3 file when the Code Conversion Workbench window was opened, the list of scripts would be empty. Once you import scripts using FmPro Migrator, click the Load Data button to populate this grid with the list of scripts.

The scripts displayed in this grid are read from the MigrationProcess.db3 file and the Status checkmarks/Completed/Not Started text are read from the ScriptsStatus.JSON file shown above.

**Note:** If the Code Conversion Workbench is running in Demo mode, only a list of demo scripts will be displayed, not the actual scripts within the MigrationProcess.db3 file.

## 3) Scripts Grid

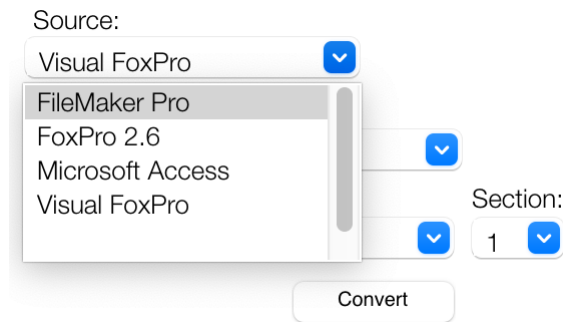
Completed: 2 of 6 Scripts					Load Data
	Status	ID	Size	Script Name	
<input type="checkbox"/>	Not Started	1	2441	pgraph.prg	
<input type="checkbox"/>	Not Started	2	4760	cgraph.prg	
<input checked="" type="checkbox"/>	Completed	3	6788	main.prg	
<input type="checkbox"/>	Not Started	4	14406	fdproc.prg	
<input type="checkbox"/>	Not Started	5	20234	datepick.prg	
<input checked="" type="checkbox"/>	Completed	6	1207	frmanimation	

The scripts grid provides a way for the developer to select specific scripts, convert them and check them off as being completed by clicking the checkmark column. The Completed ? of ? labels above the grid give a running count of the conversion status of the scripts.

Each script has an ID for reference purposes and its size in bytes and name are listed in the grid. Long script names are viewable with the scrollbar at the bottom of the grid.

Clicking any script causes it to be read from the database and displayed in the Source Script field. Clicking the script a 2nd time (to set its check/unchecked status) does not reload the script - in order to keep the contents of the Converted Script field visible.

#### 4) Source Database Type Menu



Source database types include: FileMaker Pro, FoxPro 2.6, Microsoft Access and Visual FoxPro.

Selecting the Source Database changes the title across the top of the window and re-writes the contents of the Command window.


## 5) Output Language Menu

Source:  
Visual FoxPro

Output Language:  
C#

- ABAP
- Ada
- Assembly
- Awk
- Bash
- C
- C#
- C++
- CFML
- Classic Visual Basic
- COBOL
- D
- Dart
- Delphi/Object Pascal
- Emacs Lisp
- F#
- FileMaker Pro
- Fortran
- Go
- Groovy

Section:  
1

  
Refresh


**713**

Selecting one of the over 50 output languages from the TIOBE index will rewrite the Command prompt for the selected language. Some of the languages include the specification of a framework and database name based upon the database selected on the main screen of FmPro Migrator as the output database. The text in the Command field can be manually updated to include any additional details required for the conversion.



## 6) Procedure/Function Menu

Procedure/Function:	Section:
cmdok.Click 685-749	1
BeforeDock 386-391	
AfterDock 393-399	
Undock 401-404	
Destroy 406-423	
cmdDraw.Click 425-432	
cmdDrawMode.Click 434-441	
spnPenWidth.InteractiveChange 442-443	
cboPenMode.InteractiveChange 444-445	
cboGDemo.Click 452-454	
cmdRed.CLICK 456-458	
cmdGreen.CLICK 460-462	
cmdBlue.CLICK 464-466	
cmdCustom.CLICK 468-473	
cmdErase.CLICK 475-477	
cmdClear.CLICK 479-481	
cmdDone.Click 483-486	
cboPenMode.Init 488-505	
cboGDemo.Init 507-516	
cmdok.Click 685-749	
Closing 750-757	

  
Refresh

**713**

If the main script is too large for processing by a specific machine learning model, red text "Script Too Large" will be displayed above the Source script field. If this occurs, then it is possible to break of some scripts (like Visual FoxPro) automatically into individual procedures/functions individually.

Visual FoxPro scripts often contain multiple procedures/functions. It is generally better to send smaller pieces of text to the AI models for processing, so this menu enables selecting individual procedures/functions for sending to the AI model for processing.

## 7) Section Menu

Source Script (Editable): **Script Too Large** Size: 4,759

```
***** cgraph.prg
***** C:\DS\VFP_TESTS\SOLUTION\forms\graphics\cgraph.prg
* Plot graph (Polar).
* Parameter:
* 1) cEquation (For radius) in terms of X. TYPE = Character.
* 2) nFrom. Where to stop counting for X. TYPE = Numeric.
* 3) nTo. Where to start counting for X. TYPE = Numeric.
* 4) nStepInc. Step increment. TYPE = Numeric.
* 5) nEquColor. TYPE = Numeric.
* 6) IConnect. If the previous point is connected to the current point with a line.
TYPE = Logical.
* 7) nXCenter. Point on form where x = 0. TYPE = Numeric.
```

Command (Editable):

Translate this Visual FoxPro code into C# code

Source:

Visual FoxPro

Output Language:

C#

Procedure/Function:

Closing 1-132

Section:

1

Convert

Vendor:

OpenAI

1

2

3

4

5

Some individual procedure/functions could still be too large, including the example shown here. If the "Script Too Large" script is displayed for an individual procedure/function then the Section sub-menu will also be displayed. This enables the large script to be broken up into 5 more sections (with opening/closing procedure/function text added to each section).

By the way, just because the "Script Too Large" text is displayed doesn't always mean that a script won't be processed by the AI model. You can always try sending a large script for processing, and then see if you get an error from the model.

## 8) Vendor Menu

Vendor:

OpenAI

AWS


Azure


Google

OpenAI

There are 4 AI vendors listed in the Vendor model at this time. OpenAI is the only one supported at this time, as the others are intended for future enhancements.



## 9) API Key Type

API Key:  

Model:  


By default, the Code Conversion Workbench will use the API key created by .com Solutions Inc., so "dcsi" is the default option. If you run out of AI tokens, you can generate and pay for your own API key at OpenAI.


## User API Key Field

API Key:   

To enter your own API key, select the User menu item, a new API Key field will be shown. Clicking the URL button opens a web browser to the OpenAI website where you can create an account and generate your own API key. This key gets saved with the app preferences so you only have to enter it once.

## 10) AI Model Name

Model:  

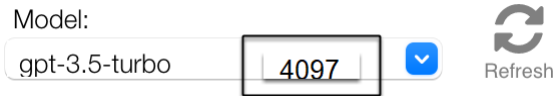
 Refresh

2,490

- gpt-3.5-turbo
- gpt-3.5-turbo-0301
- gpt-3.5-turbo-0613
- gpt-3.5-turbo-16k
- gpt-3.5-turbo-16k-0613
- gpt-4
- gpt-4-0314
- gpt-4-0613
- text-davinci-001
- text-davinci-002
- text-davinci-003

gpt-3.5-turbo will be selected as the default model. gpt-4 is also available, and generally does a significantly better job when performing conversions. But sometimes one model or the other gets overloaded so you might want to switch to another one.

## Model Token Size Tooltip



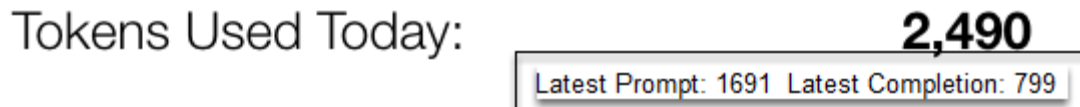
Hovering the cursor over the model menu, displays the number of tokens accepted by the selected AI model.

## 11) Model List Refresh Button



The list of available AI models is built into the app during development. You can refresh the list of models by clicking the Refresh button. The list of models displayed in the Model menu represents the models which are officially supported by the Code Conversion Workbench. At this time, there are 11 models in the list. Holding the Shift key when clicking the Refresh button displays all of the models available from the selected vendor. Selecting one of the unsupported models will generally result in an error message. Updating the list of models with the refresh button doesn't permanently save these models to the Model menu when the app is closed.

## 12) Tokens Used Today Label & Usage Tooltip



The number to the right of the Tokens Used Today text label shows the number of tokens used over the last 24 hours. Hovering the cursor over this number will display the number of tokens used during the most recent script conversion. In this example, 1691 tokens were used for the prompt and 799 tokens were used for the completed script which was returned by the model.

## Tokens Available per Day Tooltip

Tokens Used Today:

2,490

500,000 Tokens Available per Day

Hovering the cursor over the Tokens Used Today label will display the number of tokens which can be used per day for the currently selected AI model. At the present time, 500,000 tokens per day are provided for gpt3.5-turbo and the rest of the models except for gpt4. Since it currently costs 20 times as much to use the gpt4 model, the number of tokens available per day is reduced to 25,000 when gpt4 is being used. These calculations will change over time as costs change.

## 13) Output Filename Field

Output Filename:

cgraph\_Closing.cs

The name of the output script is created and saved automatically into the Converted\_Scripts folder, but it can be changed manually before pressing the Convert button. In this example the filename consists of the name of the script followed by the name of the function and ending with the C# file extension.

## Output Filename Tooltip

Output Filename:

cgraph\_Closing.cs

cgraph\_Closing\_2023-08-24\_15\_7\_46.cs

The tooltip shows the actual filename written to disk. Duplicated scripts receive a timestamp in order to preserve the original script, as you can see with this example.

## 14) Source Script Field

**Source Script (Editable):** **Script Too Large** **Size:** 4,759

```
***** cgraph.prg
***** C:\DS\VFP_TESTS\SOLUTION\forms\graphics\cgraph.prg
* Plot graph (Polar).
* Parameter:
* 1) cEquation (For radius) in terms of X. TYPE = Character.
* 2) nFrom. Where to stop counting for X. TYPE = Numeric.
* 3) nTo. Where to start counting for X. TYPE = Numeric.
* 4) nStepInc. Step increment. TYPE = Numeric.
* 5) nEquColor. TYPE = Numeric.
* 6) lConnect. If the previous point is connected to the cuurnt point with a line.
TYPE = Logical.
* 7) nXCenter. Point on form where x = 0. TYPE = Numeric.
```

Each script is read from the MigrationProcess.db3 SQLite project file when it is clicked in the scripts list grid. The script is written into the Source Script field, which can be manually edited if needed.

The size of the script in characters is displayed above the top right of the field next to the Size label. If the size of the script is estimated to be too large for the number of tokens available for the selected model, the "Script Too Large" text will be displayed above the script.

## 15) Command Field

**Command (Editable):**

Translate this Visual FoxPro code into C# code

This field provides the prompt which will be sent to the AI model. The default command for this field is created automatically when selecting the Source database type or Output Language. It is also editable with additional descriptive commands such as the programming language framework or database type.

## 16) Convert Button

Convert

The Convert button sends the Source script, Command and system message/properties to the AI model for processing. The results are written into the Converted Script field.

## 17) Converted Script Field

### Converted Script:

Here is the translated code in C#:

```
```csharp
using System;
using System.Drawing;
using System.Windows.Forms;

public class CGraph
{
    public static void PlotGraph(string cEquation, double nFrom, double nTo, double
nStepInc, int nEquColor, bool lConnect, int nXCenter, int nYCenter, bool
lAddCoords, Form frmFormName, double nEquScale)
    {
        if (string.IsNullOrEmpty(cEquation) || frmFormName == null)
        {
            MessageBox.Show("This program requires multiple parameters.");
            return;
        }

        if (nEquScale == 0)
        {
            nEquScale = 1;
        }
    }
}
```

After clicking the Convert button, the generated script is put into the Converted Script field, and also written into the Converted\_Scripts folder within the project directory.

## 18) Clipboard Icon



Clicking the clipboard icon copies the contents of the Converted Script field onto the clipboard, ready for pasting into your IDE of choice.

When FileMaker Pro is selected as the Output Language the converted script text is converted into FileMaker script XML code and placed onto the clipboard in a format which can be directly pasted into the FileMaker Pro Script Workspace window.

## Using the VFP Code Conversion Workbench

---

This chapter shows how to use the VFP Code Conversion Workbench. Before performing the code conversion with the VFP Code Conversion Workbench, you should have already imported the Visual FoxPro project into FmPro Migrator, including tables, relationships, value lists, forms/reports and scripts.

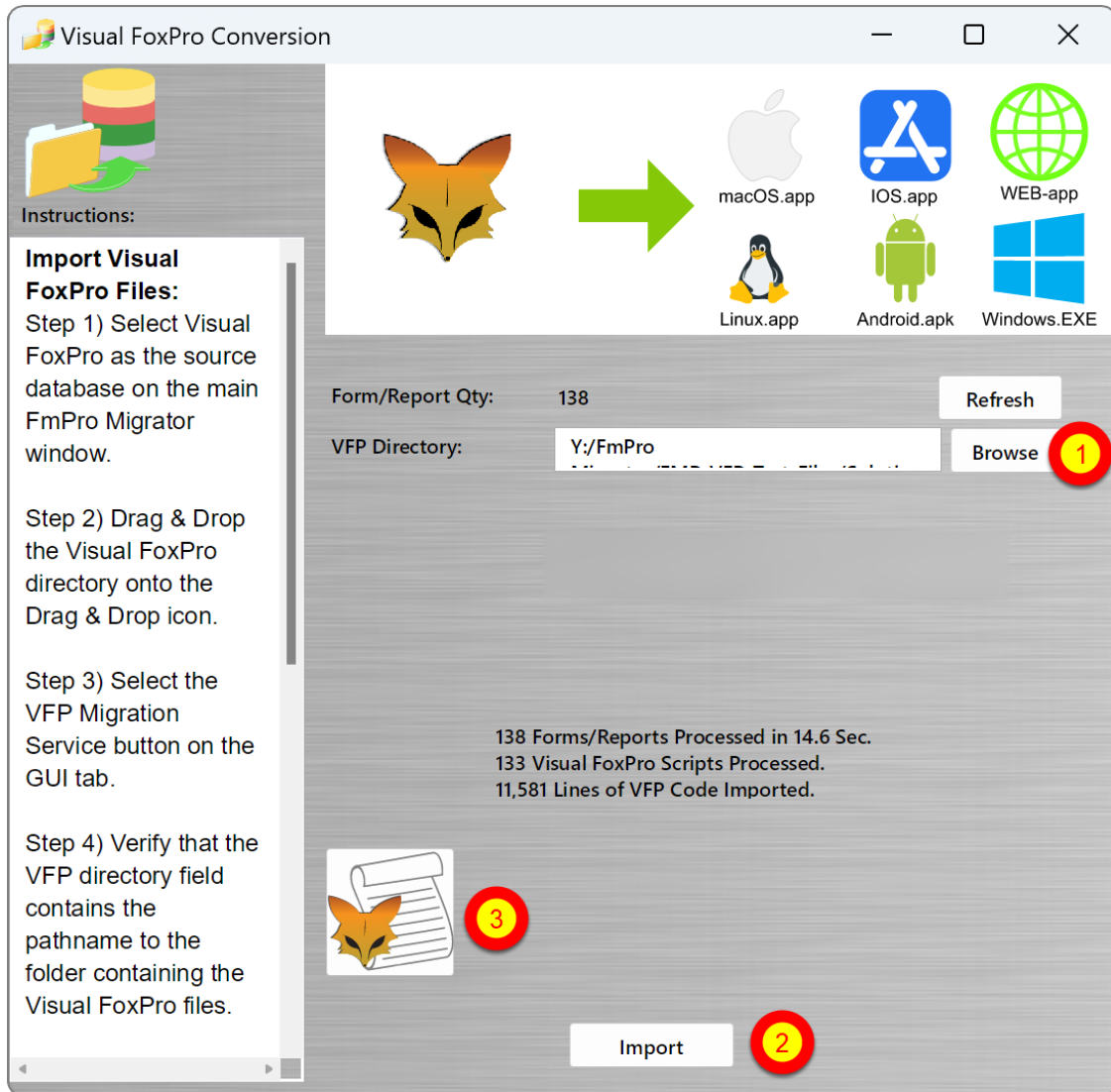
### Visual FoxPro Conversion Button - GUI Tab



Click the [Visual FoxPro Conversion](#) button on the GUI tab of the Migration Process window of FmPro Migrator.





## Visual FoxPro Conversion Window



Prior to reaching this step, you should have already selected the (1) selected the VFPEXport.DBF project folder and (2) clicked the Import button to import the VFP project into FmPro Migrator. You may have already started converting the VFP project into another development environment which would mean that you are ready (3) to click the Code Conversion Workbench button.

## VFP Code Conversion Workbench Window





# VFP Code Conversion Workbench

Completed: 2 of 6 Scripts

Load Data

Status	ID	Size	Script Name
<input type="checkbox"/> Not Started	1	2441	pgraph.prg
<input type="checkbox"/> Not Started	2	4760	cgraph.prg
<input checked="" type="checkbox"/> Completed	3	6788	main.prg
<input type="checkbox"/> Not Started	4	14406	fdproc.prg
<input type="checkbox"/> Not Started	5	20234	datepicker.prg
<input checked="" type="checkbox"/> Completed	6	1207	frmanimation

Source Script (Editable): Script Too Large Size: 4,759

```
***** cgraph.prg
***** C:\DS\VFP_TESTS\SOLUTION\forms\graphics\cgraph.prg
* Plot graph (Polar).
* Parameter:
* 1) cEquation (For radius) in terms of X. TYPE = Character.
* 2) nFrom. Where to stop counting for X. TYPE = Numeric.
* 3) nTo. Where to start counting for X. TYPE = Numeric.
* 4) nStepInc. Step increment. TYPE = Numeric.
* 5) nEquColor. TYPE = Numeric.
* 6) lConnect. If the previous point is connected to the current point with a line.
TYPE = Logical.
* 7) nXCenter. Point on form where x = 0. TYPE = Numeric.
```

Command (Editable):  
Translate this Visual FoxPro code into C# code

Converted Script:  
Here is the translated code in C#:  

```
'''csharp
using System;
using System.Drawing;
using System.Windows.Forms;

public class CGraph
{
    public static void PlotGraph(string cEquation, double nFrom, double nTo, double
nStepInc, int nEquColor, bool lConnect, int nXCenter, int nYCenter, bool
lAddCoords, Form frmFormName, double nEquScale)
    {
        if (string.IsNullOrEmpty(cEquation) || frmFormName == null)
        {
            MessageBox.Show("This program requires multiple parameters.");
            return;
        }

        if (nEquScale == 0)
        {
            nEquScale = 1;
        }
    }
}
```

Source: Visual FoxPro

Output Language: C#

Procedure/Function: Closing 1-132

Section: 1

Convert

Vendor: OpenAI

API Key: dcsi

Model: gpt-3.5-turbo

Tokens Used Today: 5,071

Output Filename: cgraph\_Closing.cs

Need Help?

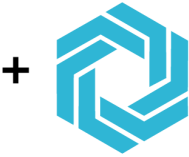

By default, the VFP Code Conversion Workbench window opens with the (1) scripts listed in the grid, (2) Visual FoxPro selected as the Source Database Type and (3) C# selected as the destination language.

Depending upon the destination database selected on the main FmPro Migrator window, you could also see FileMaker Pro or Microsoft Access VBA selected as the Output Language.

Just because the options are selected a certain way when the window opens, doesn't mean you can't change them.

You can easily change the default settings on this window for Output Language, Vendor, API Key Type, or AI Model - and the changes will be saved into the project or application preferences for FmPro Migrator.

## Converting FoxPro 2.6 to Visual FoxPro



# FoxPro 2.6 Code Conversion Workbench

Completed: 2 of 6 Scripts

Status	ID	Size	Script Name
<input checked="" type="checkbox"/> Completed	1	2441	pgraph.prg
<input checked="" type="checkbox"/> Completed	2	4760	cgraph.prg
<input checked="" type="checkbox"/> Completed	3	6788	main.prg
<input type="checkbox"/> Not Started	4	14406	fdproc.prg
<input type="checkbox"/> Not Started	5	20234	datepicker.prg
<input type="checkbox"/> Not Started	6	1207	frmanimation

Source Script (Editable):

Size: 643

```
set talk off
set stat off
set scor off
set cent on
set date brit
do while .t.
  clear
  @5,20 to 19,45 doub
  @6,25 say "MAIN MENU"
  @7,21 to 7,44 doub
  k=0
  @ 9,25 prompt "DATA ENTRY"
```

Command (Editable):

Translate this FoxPro 2.6 code into Visual FoxPro PRG code to build an .SCX form to represent the user interface from the FoxPro SAY commands.

Converted Script:

```
PRG
SET TALK OFF
SET STAT OFF
SET SCORE OFF
SET CENT ON
SET DATE BRIT

CREATE FORM frmMainMenu

WITH frmMainMenu
  ADD OBJECT lblMainMenu TO 5,20 WIDTH 19 HEIGHT 25 ;
  VALUE "MAIN MENU"
  ADD OBJECT ineMainMenu TO 7,21 WIDTH 23 HEIGHT 1 ;
  ADD OBJECT cmdDataEntry TO 9,25 WIDTH 15 HEIGHT 1 ;
  VALUE "DATA ENTRY"
  ADD OBJECT cmdReportPrinting TO 11,25 WIDTH 15 HEIGHT 1 ;
  VALUE "REPORT PRINTING"
  ADD OBJECT cmdProcess TO 13,25 WIDTH 15 HEIGHT 1 ;
  VALUE "PROCESS"
  ADD OBJECT cmdExit TO 15,25 WIDTH 15 HEIGHT 1 ;
  VALUE "EXIT"
  ADD OBJECT cmdQuitSystem TO 17,25 WIDTH 15 HEIGHT 1 ;
  VALUE "QUIT TO SYSTEM"
ENDWITH
```

Source: FoxPro 2.6

Output Language: Visual FoxPro

Convert

Vendor: OpenAI

API Key: dcsi

Model: text-davinci-003

Tokens Used Today: 1,502

Output Filename: fp26\_mainmenu\_proc1.prg

Need Help?

Selecting FoxPro 2.6 as the Source Database type enables the conversion of FoxPro 2.6 code into Visual FoxPro code, including the creation of forms from the FoxPro 2.6 command line code. Performing this conversion makes use of the text-davinci-003 model. You will also need to break up the size of the script text manually, if there aren't any procedures/functions.

In order to prepare the FoxPro 2.6 project for import into FmPro Migrator, perform the following steps:

- 1) Install Visual FoxPro 9.
- 2) Make a copy of the FoxPro 2.6 project.
- 3) Open and convert the copied project in Visual FoxPro 9. This will upgrade all of the files including the DBF files. This is why you want to work with a copy of the project - because the DBF files won't be readable in FoxPro after conversion. And you might need to continue developing the existing FoxPro app while performing your conversion.
- 4) Once the FoxPro 2.6 project has been converted into Visual FoxPro 9, create a DBC and add all of the DBF files to the DBC.

5) Manually add 1 empty form to the Visual FoxPro 9 project.

Now you are ready to follow the existing PDF manual showing how to import the project into FmPro Migrator for conversion.

### VFP Code Conversion Workbench Demo



## VFP Code Conversion Workbench Demo

If you forgot to enter your license key or didn't have a license for the AI Accelerated version of FmPro Migrator, then the VFP Code Conversion Workbench will open in Demo mode.

Apre-converted set of sample scripts will be displayed in the grid instead of the scripts from your own Visual FoxPro project.

Clicking the Convert button will place the pre-converted script into the Converted Script field.