

Visual FoxPro to Servoy Conversion Procedure



.com Solutions Inc.

1	Step 1 - Get Info	
1.1	Visual FoxPro Conversion - Step 1 - Get Info	4
2	Step 2 - Create Table	
2.1	Visual FoxPro Conversion - Step 2 - Create Table	18
3	Step 3 - Transfer Data	
3.1	Visual FoxPro Conversion - Step 3 - Transfer Data	23
4	Import MetaData	
4.1	Visual FoxPro Conversion - Import MetaData	30
5	Visual FoxPro .MPR Menu Conversion	
5.1	Visual FoxPro Menu Converter - Servoy	39
6	Visual FoxPro .PRG Code Conversion	
6.1	Visual FoxPro Code Converter - Servoy	45
7	Servoy Conversion	
7.1	Create Servoy Eclipse Project	58
7.2	Enter Servoy Project Parameters	60
7.3	Servoy Migration - Migrate Button	63
7.4	Servoy Migration - Refresh Project	64
7.5	Post Migration Tasks - Visual FoxPro to Servoy Migration	66

Step 1 - Get Info

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.

Document Version 7

Updated for FmPro Migrator Developer Edition 5.97

Updated with Visual FoxPro Menu Converter Info

Updated with Visual FoxPro Code Converter Info

1/12/2011

Pre-Migration Tasks

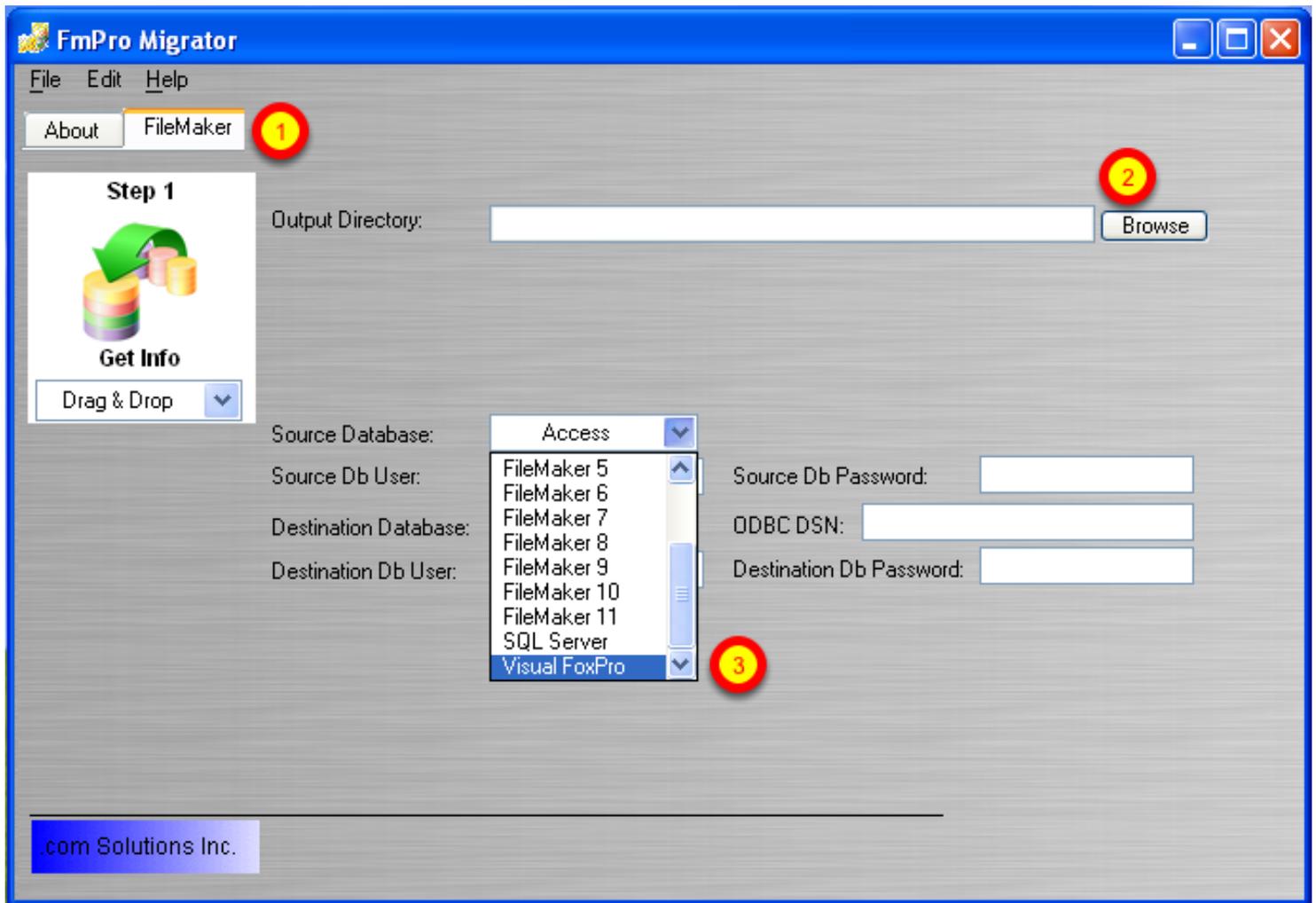
Before starting the Visual FoxPro Conversion process:

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.

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

Step 1

Output Directory: C:/ds/VFP_Tests/Solution_VFP/ Browse

Source Database: Visual FoxPro

Destination Database: **1** SQL Server ODBC DSN: sqlserver_dsn **2**

Dest. DB Hostname: **3** localhost **4**

Dest. DB Port: 3333

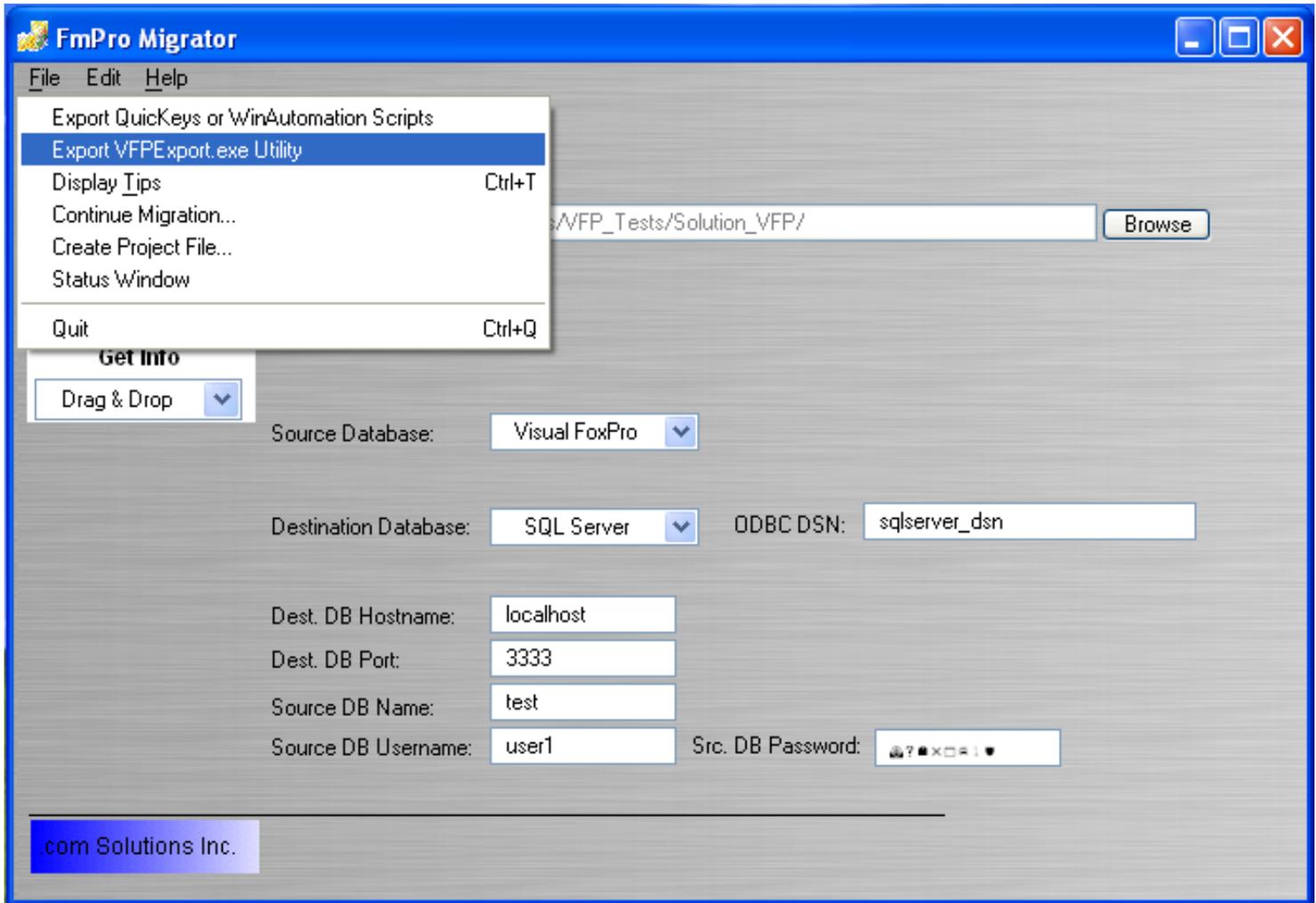
Source DB Name: **5** test

Source DB Username: user1 **6** DB Password: **7**

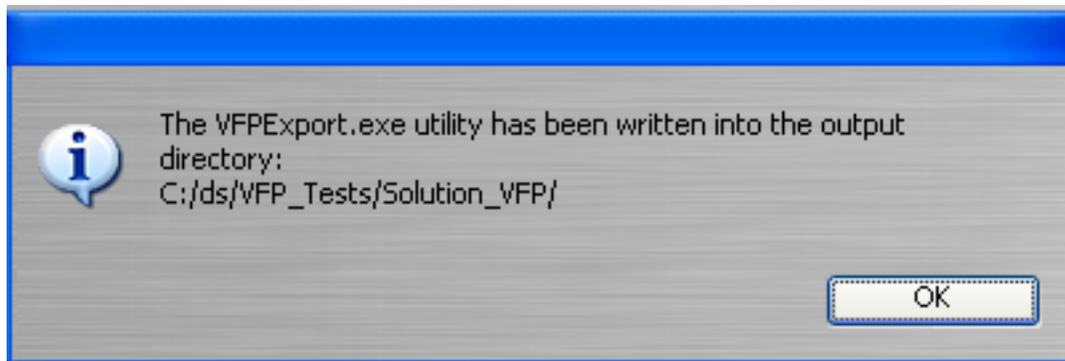
.com Solutions Inc.

(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 .PGX 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 .PGX file. The resulting exported data is stored within the VFPEXport.DBF database file.



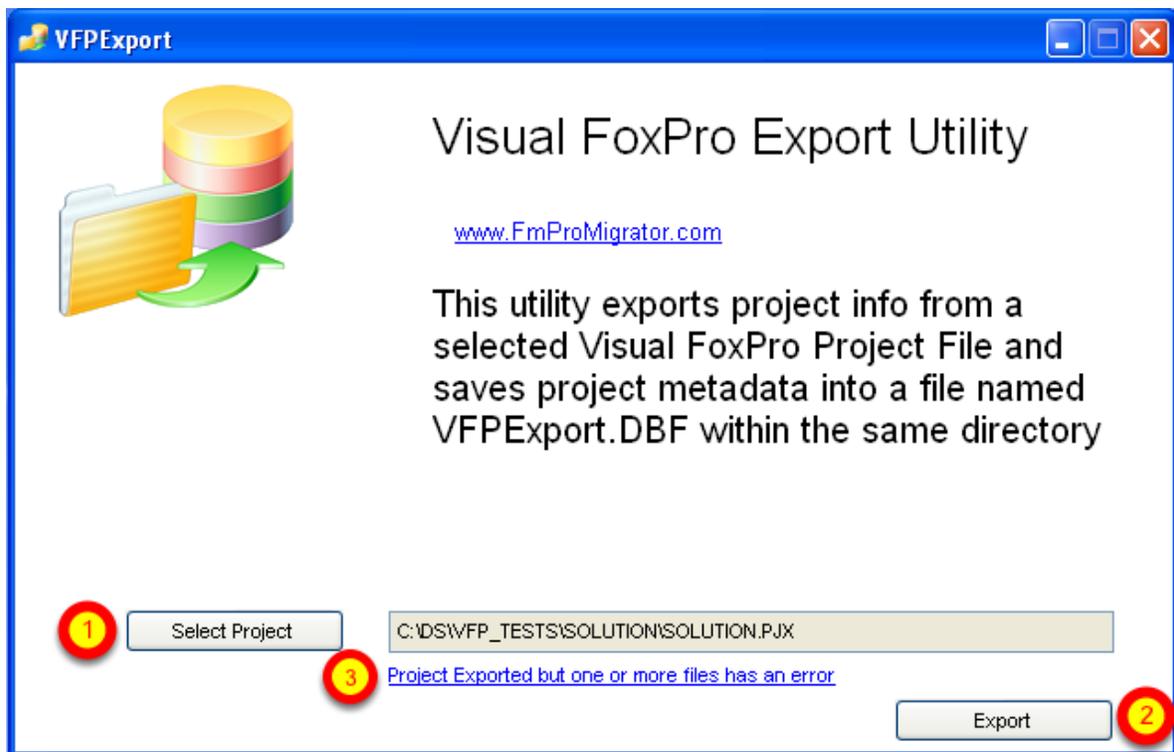
The VFPEXport utility will be saved into the FmPro Migrator Output Directory.

VFPEXport Utility - Static MetaData Export Process



FmPro Migrator uses a 2 step process to import the metadata from within Visual FoxPro projects. Through 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



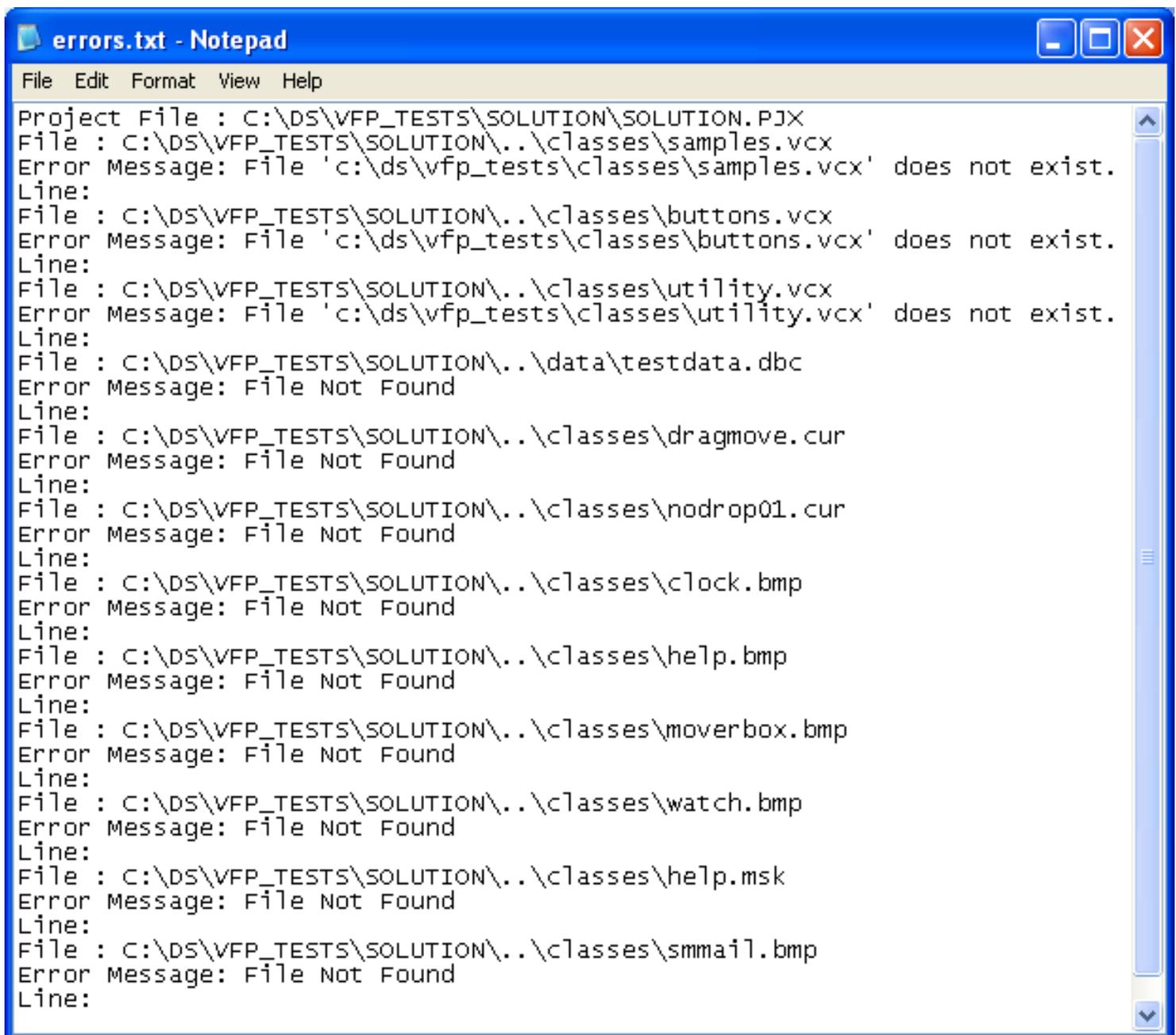
Launch the VFPEXport.exe utility, (1) Select the Visual FoxPro project file for the project to be converted, (2) click the Export button.

(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.

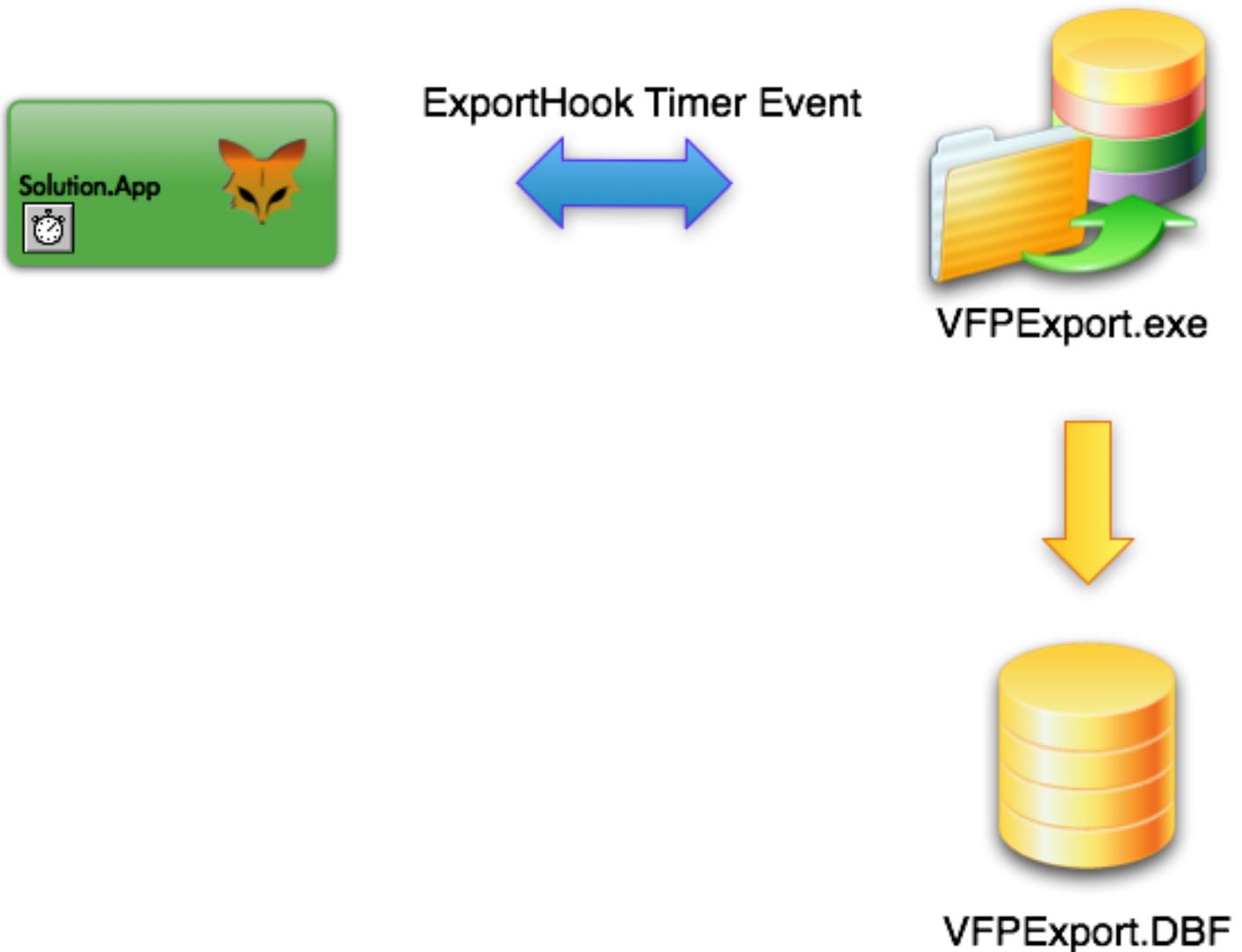
VFPEXport Error Log File



```
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.

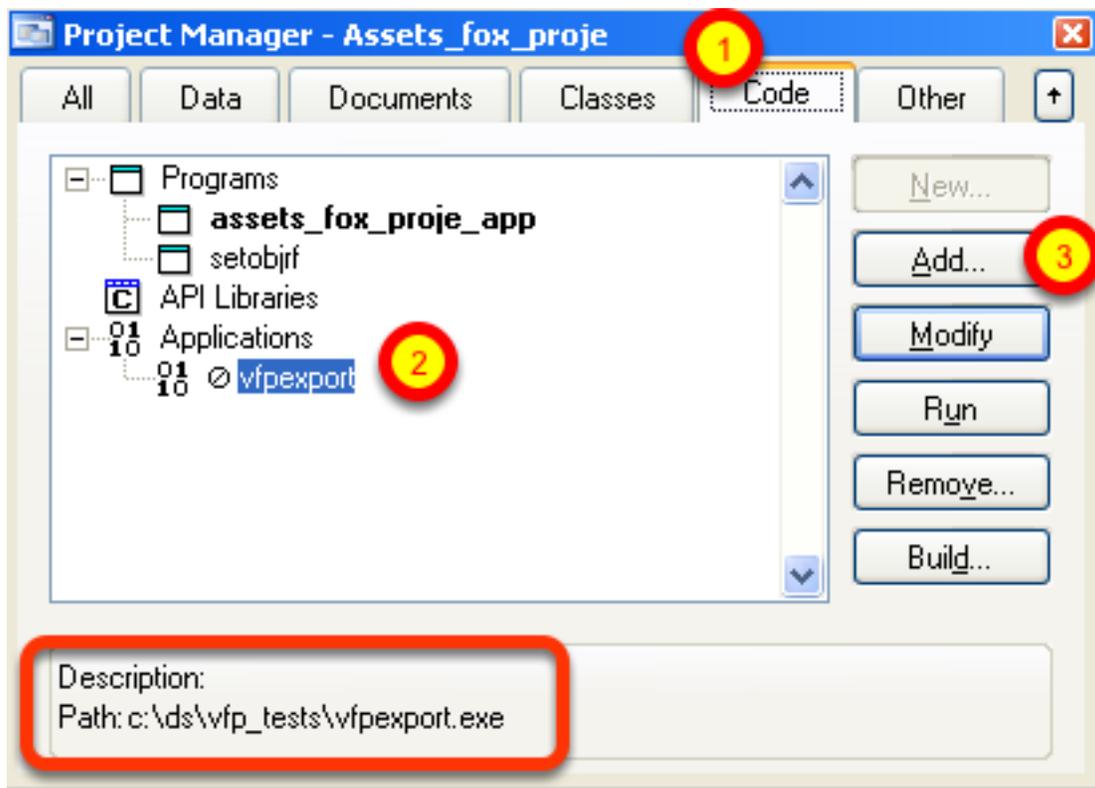
VFPEXport Utility - Runtime MetaData Export Process



Since most Visual FoxPro projects contain VCX forms and objects having properties set only at runtime, an additional AppHook export process is also used to export project metadata. The VFPEXport.exe utility is added to the startup code of the Visual FoxPro application. The Visual FoxPro project is compiled and then run as a user would use the application to open and close all of the forms within the application. One second after each form is instantiated, the ExportHook timer event fires and captures metadata for all of the objects on the current form/report. Once the Visual FoxPro application is closed, the new data is merged with the existing form/report data within the VFPEXport.DBF file. By capturing this information during runtime, all inherited object properties will have been set by the application code, thus insuring that an accurate representation of the objects has been exported. After the capture process has been completed, the records within the VFPEXport.DBF table can be modified or even deleted in order to remove obsolete or unused form records.

Note: When FmPro Migrator reads the records within the VFPEXport table, it first starts by looking for each of the form and report records. Then it looks for all of the objects associated with the form or report which is being processed.

Adding ExportHook to Visual FoxPro Project



- 1) In the Visual FoxPro project it is necessary to set a reference to the VFPEExport.exe executable file. This is done in the Code -> Applications tab of the Project Manager.
- 2) Add the following code to the project's Startup (main PRG or startup form's Init / Load method) code. Replace the pathname to the VFPEExport.exe app with the actual path to the executable on your computer. Also, replace the path to the VFPEExport.DBF file with the actual path to this database file, usually located at the top-level of your project within the directory which contains your project .PJX file.

PUBLIC oFormHook

```
oFormHook = NEWOBJECT("ExportHook", "bzAppHook.VCX", "C:\Project\VFPEExport.exe",  
"C:\Project\VFPEExport.dbf")
```

Try

```
Do StartApplication
```

Catch (...)

```
...
```

Finally

```
Release oFormHook
```

End

- 3) Build and run the Visual FoxPro project normally.
- 4) Open each of the forms in your Visual FoxPro application, just as a user would use your application. During runtime, the ExportHook event will fire and automatically store the properties of every object on each new form which is opened.

5) When the VisualFoxPro application closes and the ExportHook is released, the VFPEExport.exe utility will merge the captured data with data already extracted by previous run of the utility.

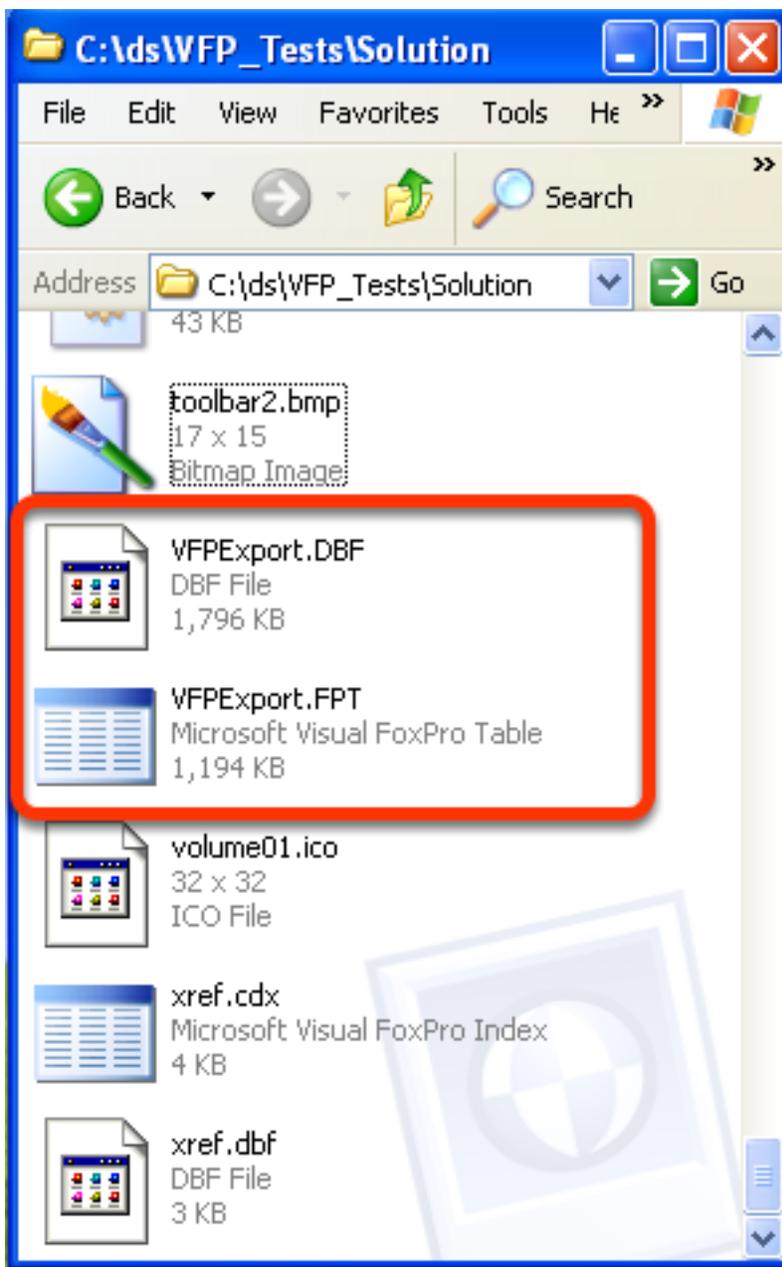
Usage Notes:

- 1) If the VFPEExport.dbf file is found in the root directory of target application, it is opened and for every form captured, the corresponding record is searched within the VFPEExport.DBF file. If the form was found, the properties are updated with the properties gathered at runtime.
- 2) If the form was not found, then a new record is created and captured data is saved into the VFPEExport.DBF file.
- 3) This process works for both stand-alone SCX forms as well as VCX forms defined within libraries.
- 4) If the VFPEExport.DBF file is not found, the captured data is exported to a different file named VFPEExport_extra.DBF.
- 5) If any error occurs during the properties merging process, the captured data is exported into the VFPEExport_extra.DBF file.

Since it is possible that the VFPEExport.DBF file might not be found during runtime, it is recommended that a small sample of forms be tested in order to verify that the data is getting written into the VFPEExport.DBF file. FmPro Migrator is only looking for the VFPEExport.DBF file during its processing, and does not automatically merge data in from the VFPEExport_extra.DBF file. This is because the VFPEExport_extra.DBF file doesn't contain all of the metadata for the entire project, and FmPro Migrator may not be able to predict where the VFPEExport_extra.DBF file is located on disk.

The Visual FoxPro developer can manually copy the records from the VFPEExport_extra.DBF file into the original VFPEExport.DBF file - but it will be easier to just make sure that the process is working properly to avoid performing this extra manual processing step.

VFPEXport.DBF/FPT Files



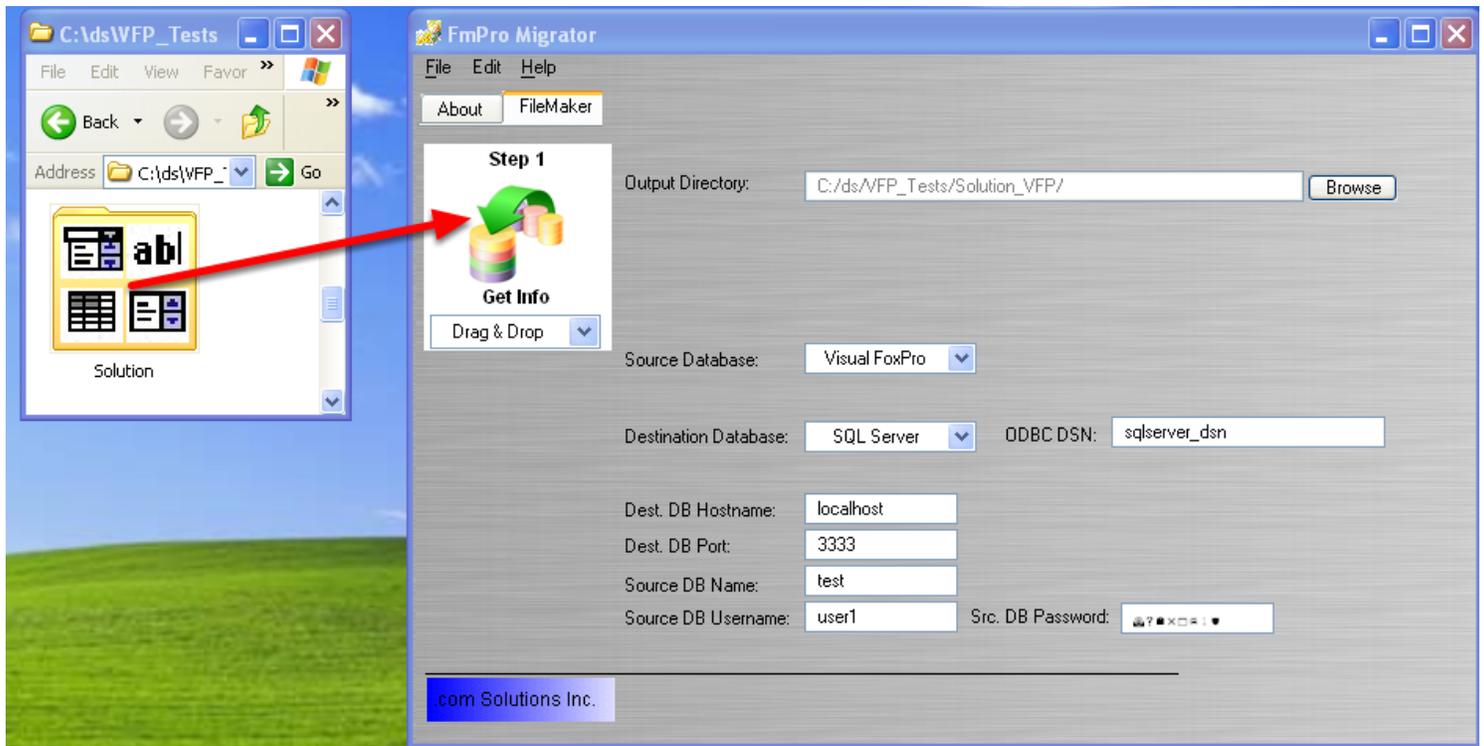
As a result of running the VFPEXport.exe utility, the VFPEXport.DBF/FPT files will be found within the Visual FoxPro project's directory. These files provide the Visual FoxPro project metadata which will be used by FmPro Migrator for the remaining steps of the conversion process.

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

123 abc abc Varchar Text

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: Table: customer
 Cascade Create: Cascade Delete:
 Relationship SQL Code:

Right Table: Table: orders
 Cascade Create: Cascade Delete:

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

Visual FoxPro Conversion - Step 2 - Create Table

Step 2 - Create Table

The screenshot shows the 'Migration Process' application window. The 'Tables' tab is active, displaying a list of 7 tables. The 'Original Table' column has 'dvdprice' selected, indicated by a red circle with the number '1'. Below the table list, the 'Fields' section shows 3 fields: 'id', 'price', and 'listprice'. At the bottom of the interface, two step cards are visible: 'Step 2: Create Table' (highlighted with a red box and a red circle with the number '2') and 'Step 3: Transfer Data'. The 'Create Table' card shows a green plus sign over a table icon and the text 'Not Started'. The 'Transfer Data' card shows a database icon with a green arrow and the text 'Not Started'. On the left side, there are instructions for Step 1 and Step 2.

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	

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

Step 2
Create Table
Not Started

Step 3
Transfer Data
Not Started

com Solutions Inc.

(1) Click on any table, (2) click on the Create Table button.

Create Table - Troubleshooting



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:

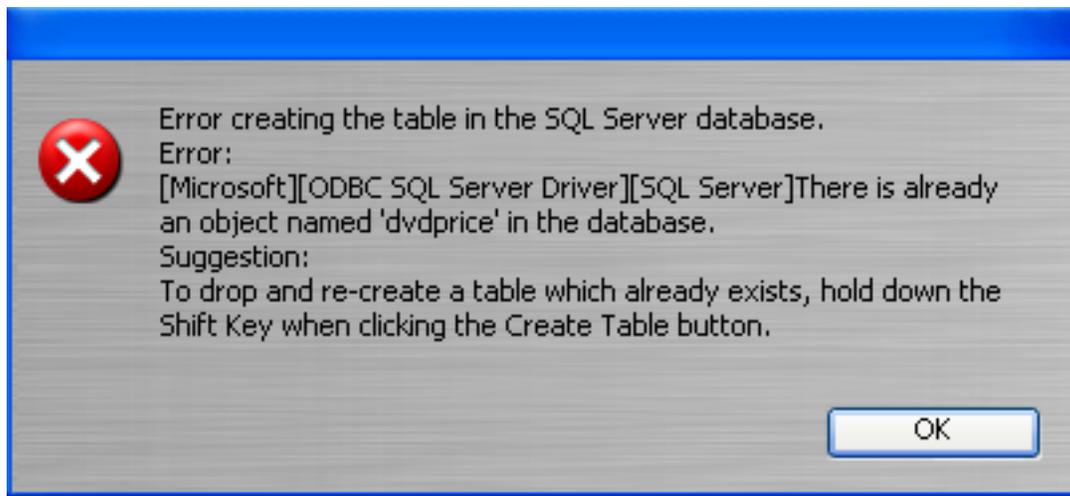
```
CREATE TABLE [Assets].[user1].[dvdprice]
(
  id
  VARCHAR(25)                NOT NULL,
  price
  DECIMAL(18,4)              NOT NULL,
  listprice
  DECIMAL(18,4)              NOT NULL
)
```

Post Table Creation SQL Code:

Drop Table
Execute SQL
Save

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' application window. The 'Tables' tab is active, displaying a table with 7 rows and 6 columns: Original Table, New Table, Source, Destination, Fields, and Record. The 'Fields' tab is also active, showing 3 fields with columns: ID, Original Fieldname, PK, Auto-Increment, Next SN, and New Fieldname. The 'Instructions' pane on the left contains two steps. Step 1 describes selecting the VFPEXport.exe utility. Step 2 describes launching the utility and clicking the Export button. Step 3, 'Transfer Data', is highlighted with a red border and shows a 'Not Started' status. The 'Create Table' step is shown as 'Completed'.

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 3:
Transfer Data
Not Started

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	

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

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' application window. The 'Tables' tab is active, displaying a table mapping interface. Below it, the 'Fields' tab shows a mapping of fields from the original table to the new table. A red box highlights the 'Column Change' buttons for the 'id' field, showing options for '123', 'abc', 'abc', and 'Varchar Text'.

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	

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, abc, Varchar Text

Step 2: Create Table - Completed

Step 3: Transfer Data - Not Started

© 2008 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.

Troubleshooting - Table Details Window

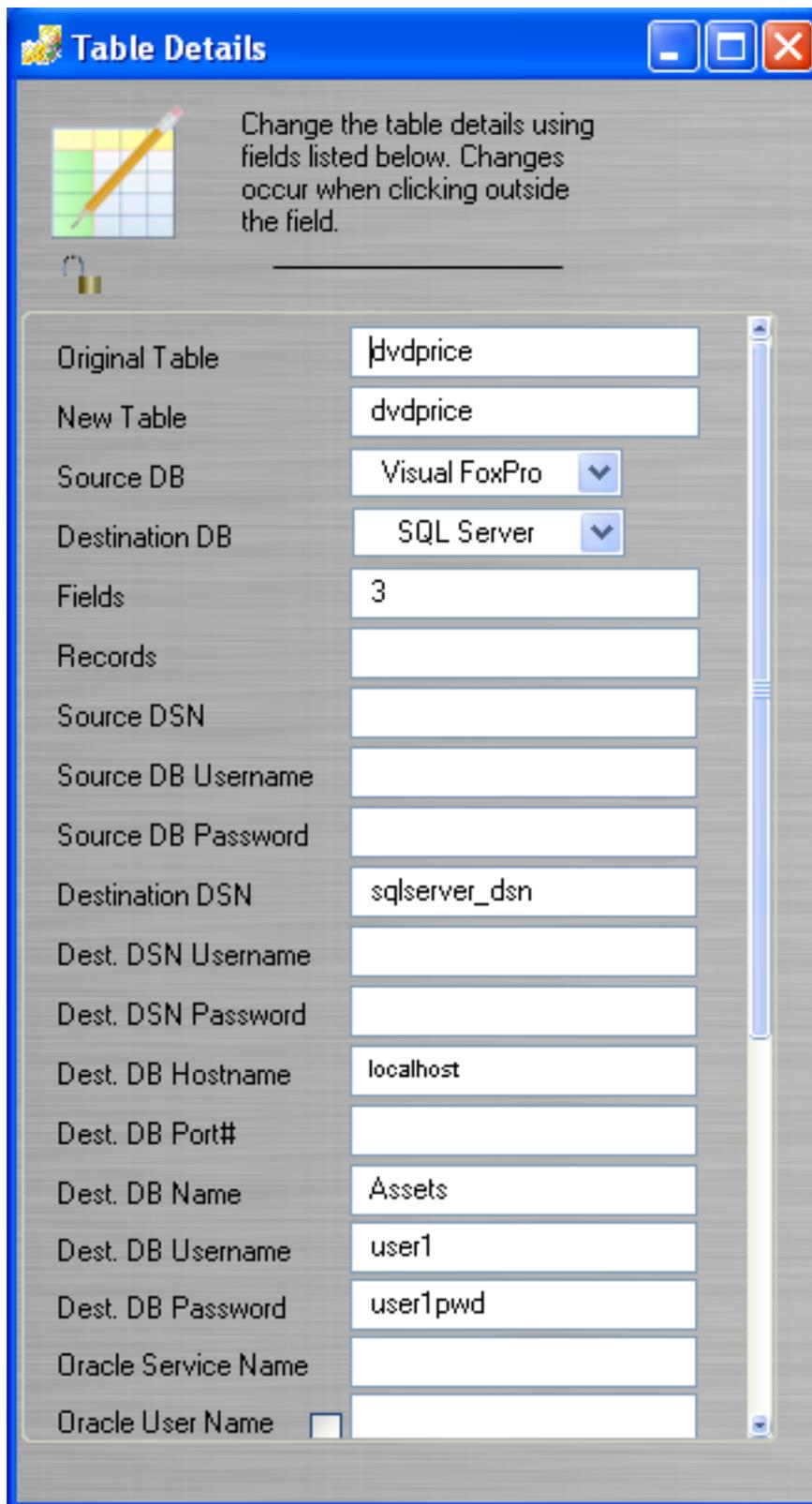


Table Details

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

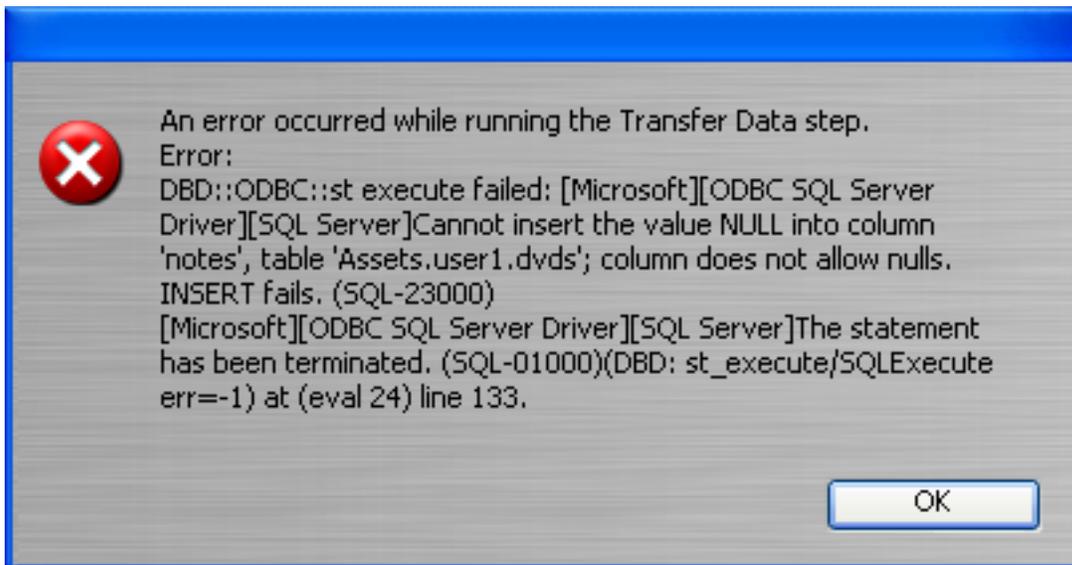
Dest. DB Password: user1pwd

Oracle Service Name:

Oracle User Name:

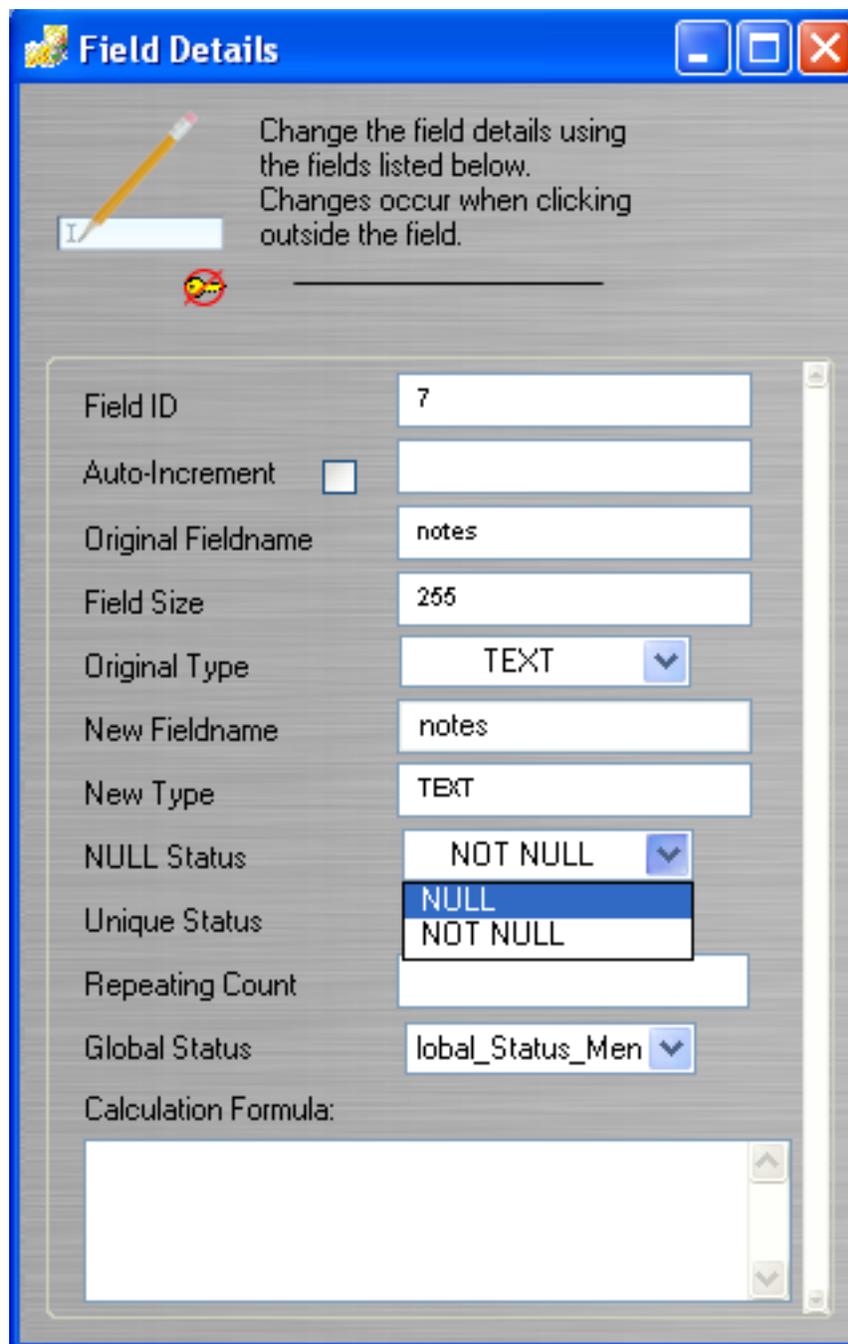
The Table Details window can be displayed by double-clicking on a table in the list of tables. Clicking a different table with the Table Details window displayed will automatically update the info in the Table Details window with the info for the table which has been clicked.

Troubleshooting - NOT NULL Column



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.

Troubleshooting - Field Details Window



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' application window. The 'Tables' tab is active, displaying a list of 7 tables. A red box highlights the 'XLS' report button in the toolbar. The interface also shows instructions for two steps, a field summary table, and progress indicators for 'Step 2: Create Table' and 'Step 3: Transfer Data', both marked as 'Completed'.

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: 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

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.

Visual FoxPro to Servoy Conversion Procedure - 28

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 Service Button

The screenshot shows the 'Migration Process' application window. The 'GUI' tab is selected and highlighted with a red circle containing the number '1'. The window contains a sidebar with instructions for converting layouts to HTML and XML, and a main area with a grid of service buttons. The 'Visual FoxPro Conversion Service' button is highlighted with a red circle containing the number '2'. The window title bar includes standard Windows window controls (minimize, maximize, close) and a red circle with the number '1' next to the 'GUI' tab.

Instructions:

Convert Layouts to HTML:
Click the Convert Layouts to HTML button to convert each FileMaker Pro layout into an HTML file. Each file includes fields, text labels, images, graphic objects and JavaScript code for implementing portals and pop-up calendar pickers.

Export Layout XML:
Click the Export Layouts as XML button to export the layout definition XML for all layouts as text files. Each layout is exported as a text file within a folder named Layout_XML_Files within the output directory.

Access to FmPro

Services

- Convert Layouts to HTML
- Export Layouts as XML
- Access to FmPro Migration Service
- FmPro to Access Migration Service
- Servoy Migration Service
- LiveCode Conversion
- PHP to LiveCode Conversion
- Visual FoxPro Conversion Service

.com Solutions Inc.

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

Visual FoxPro Conversion Service Window - Features

Visual FoxPro Conversion Service

Instructions:

Import Visual FoxPro Files:

Step 1) Select Visual FoxPro as the source database on the main FmPro Migrator window.

Step 2) Drag & Drop the Visual FoxPro directory onto the Drag & Drop icon.

Step 3) Select the VFP Migration Service button on the GUI tab.

Step 4) Verify that the VFP directory field contains the pathname to the folder containing the Visual FoxPro files.

Form/Report Qty: Refresh (2)

VFP Directory: Browse (1)

Processing Type: Demo (3)

QUOTE \$

Import (4)

(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) Use Demo mode or order a License Key for the quantity of Forms/Reports which need to be processed.

To enter the Visual FoxPro license key, select "Licensed" from the Processing Type menu, copy the license key to the clipboard from your email program, click on the clipboard icon to the left of the license key field. The license key will be copied from the clipboard and validated.

(4) Click the Import button to convert and import and the Visual FoxPro project metadata into FmPro

Migrator.

Note: Without a Visual FoxPro Conversion license key, up to 5 Forms/Reports will be converted in Demo mode.

Visual FoxPro Conversion Service Window - Refresh Button



After selecting the Visual FoxPro project directory, clicking the Refresh button reads the number of Forms/Reports from the VFPEXport.db3 SQLite database and displays the info on the Visual FoxPro Conversion window.

Visual FoxPro Conversion Service Window - Import Button

The screenshot shows the 'Visual FoxPro Conversion Service' window. On the left, there is an 'Instructions' pane with four steps: 1) Select Visual FoxPro as the source database on the main FmPro Migrator window. 2) Drag & Drop the Visual FoxPro directory onto the Drag & Drop icon. 3) Select the VFP Migration Service button on the GUI tab. 4) Verify that the VFP directory field contains the pathname to the folder containing the Visual FoxPro files.

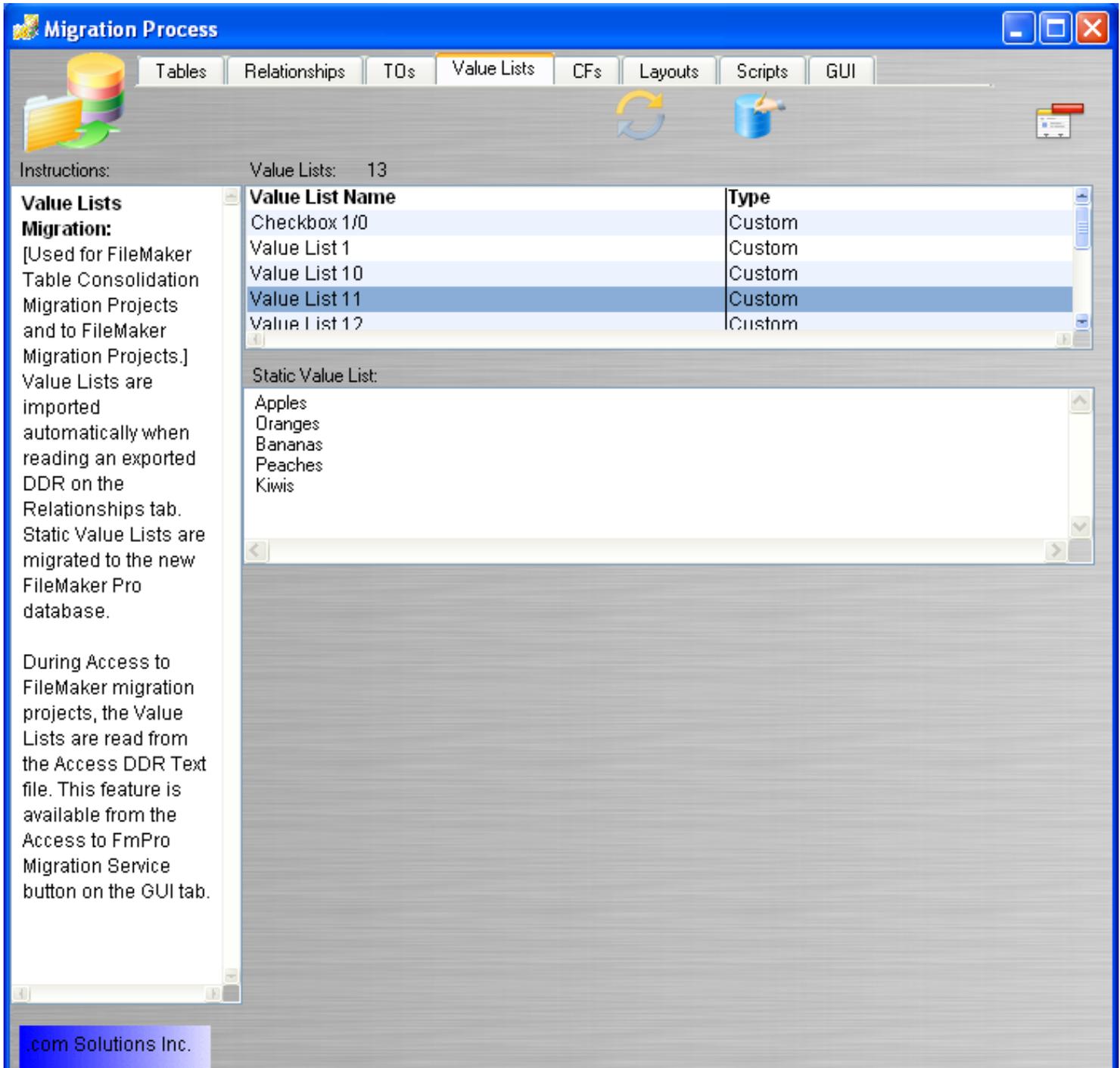
The main window area features a fox logo, a blue arrow pointing to a database cylinder, a smartphone, and a tablet. Below this, there are input fields: 'Form/Report Qty: 87' with a 'Refresh' button; 'VFP Directory: C:/ds/VFP_Tests/Solution' with a 'Browse' button; 'Processing Type: Licensed' with a dropdown arrow; and 'License Key:' with a clipboard icon and a text field.

A red-bordered box highlights the following text: **Valid License Key**
Valid Until 11/12/2011 (87 Forms/Reports)
87 Forms/Reports Processed in 16.6 Sec.
1692 Visual FoxPro Scripts Processed.

At the bottom, there is an 'Import' button with a red circle containing the number '1' next to it. A red circle containing the number '2' is also present next to the success message box.

(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



The screenshot shows the 'Migration Process' window with the 'Value Lists' tab selected. The window title is 'Migration Process' and it has standard Windows window controls. The top navigation bar includes 'Tables', 'Relationships', 'TOs', 'Value Lists', 'CFs', 'Layouts', 'Scripts', and 'GUI'. Below the navigation bar, there are icons for a folder, a refresh button, and a database icon. The main area is divided into two sections: 'Instructions' on the left and a list of 'Value Lists' on the right. The 'Instructions' section contains text explaining the migration of Value Lists and Static Value Lists. The 'Value Lists' section shows a table with columns 'Value List Name' and 'Type'. The table lists five Value Lists, all of which are 'Custom'. Below the table, there is a section for 'Static Value List' which displays a list of fruit names: Apples, Oranges, Bananas, Peaches, and Kiwis. At the bottom left of the window, there is a small blue box with the text '.com Solutions Inc.'

Value Lists
Migration:
[Used for FileMaker Table Consolidation Migration Projects and to FileMaker Migration Projects.]
Value Lists are imported automatically when reading an exported DDR on the Relationships tab. Static Value Lists are migrated to the new FileMaker Pro database.

During Access to FileMaker migration projects, the Value Lists are read from the Access DDR Text file. This feature is available from the Access to FmPro Migration Service button on the GUI tab.

Value List Name	Type
Checkbox 1/0	Custom
Value List 1	Custom
Value List 10	Custom
Value List 11	Custom
Value List 12	Custom

Static Value List:

- Apples
- Oranges
- Bananas
- Peaches
- Kiwis

.com Solutions Inc.

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

The screenshot shows the 'Migration Process' application window. The 'Layouts' tab is selected, displaying a list of layouts for the 'products' table. The 'dyngrid' layout is highlighted. Below the list, the 'Portals' section shows 'Portal1' with a fill method of 'On Record Advance'. A 'Layout Image' preview shows a yellow rectangle labeled 'Portal1' within a larger white rectangle.

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	Portals
dddrop	490,320	0	0
dsession	749,273	4	0
dyngrid	558,498	13	1
editbox	519,498	4	0
fillmenu	536,382	3	0

Portal Name	Fill Method	Field Titles
Portal1	On Record Advance	

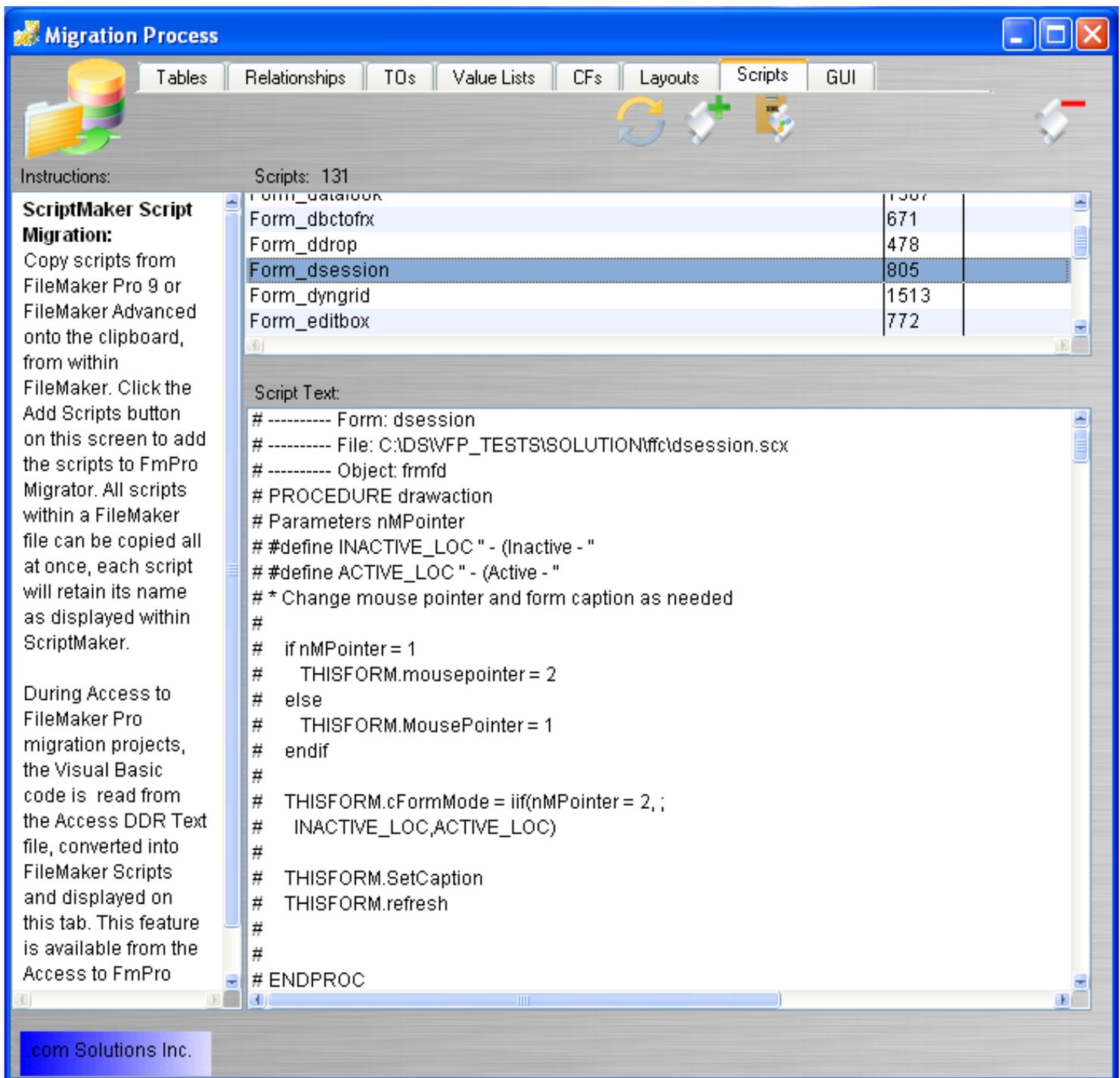
Layout Image

Portal1

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



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	Line Count
Form_dataform	1307
Form_dbctofix	671
Form_ddrop	478
Form_dsession	805
Form_dyngrid	1513
Form_editbox	772

Script Text:

```
# ----- Form: dsession
# ----- File: C:\DSWFP_TESTS\SOLUTION\ifc\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.

License Key Note: The Visual FoxPro Conversion license key used during the Visual FoxPro import process, can also be used to unlock all of the other conversion targets on the GUI tab of the Migration Process window, in order to complete the conversion process.

Visual FoxPro .MPR Menu Conversion

Visual FoxPro Menu Converter - Servoy

FmPro Migrator Developer Edition includes the MacOS X and Windows versions of the Visual FoxPro Menu Converter as a compiled stand-alone application. This application converts FoxPro generated menu .MPR files into Servoy JavaScript code.

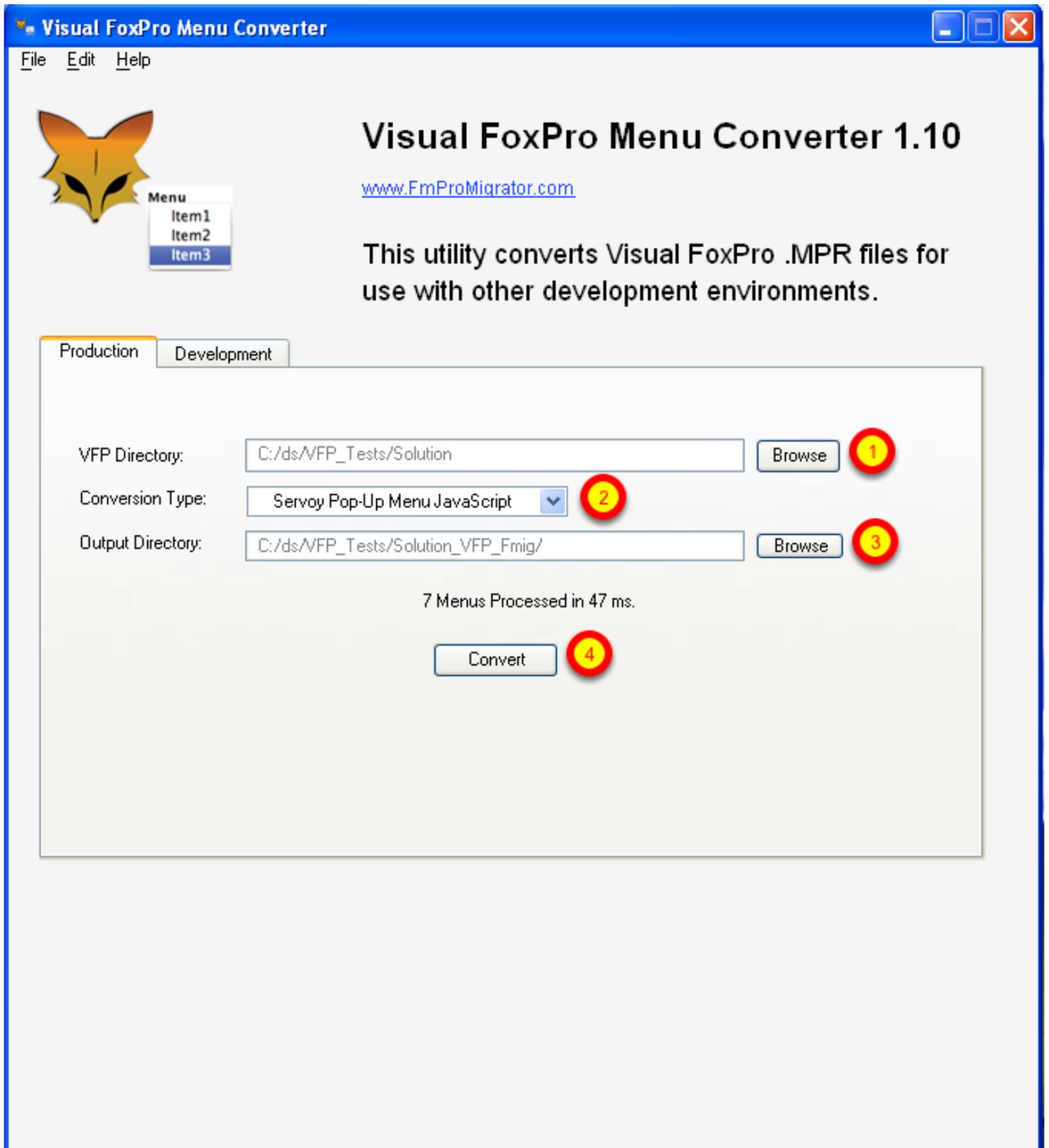
Visual FoxPro Menu Converter 1.10

1/1/2011

Introduction

The Visual FoxPro Menu Converter included with FmPro Migrator Developer Edition reads the GENMENU code from FoxPro .MPR files and converts the code into Servoy JavaScript code. Menus can be converted individually, as shown in the demo below, or can be converted for an entire Visual FoxPro project all at once. The Conversion Type menu provides two conversion options. Visual FoxPro menu STYLE parameters for menu items are converted into embedded HTML tags, supporting the style options: Bold, Italic, Underline, and Strikeout.

ON SELECTION statement procedure calls are converted into empty Servoy JavaScript functions to enable quick testing of the generated menu code.



The Production tab will read all of the Menu records from the VFPEXport.db3 SQLite database file which was created at Step 1 of the conversion process. To perform a production conversion, (1) select the Visual FoxPro project folder containing the VFPEXport.db3 file, (2) select the type of JavaScript

menu code to generate, (3) select the output directory where the generated JavaScript file will be written, (4) click the Convert button.

Development Conversion Tab

Visual FoxPro Menu Converter 1.10

www.FmProMigrator.com

This utility converts Visual FoxPro .MPR files for use with other development environments.

Production Development

Clear Source Code 1

Visual FoxPro Menu Source Code:

```
DEFINE MENU example BAR AT LINE 1
DEFINE PAD convpad OF example PROMPT '\<Conversions' COLOR SCHEME 3;
KEY ALT+C, "
DEFINE PAD cardpad OF example PROMPT 'Card \<Info' COLOR SCHEME 3;
KEY ALT+I,"
ON PAD convpad OF example ACTIVATE POPUP conversion
ON PAD cardpad OF example ACTIVATE POPUP cardinfo
DEFINE POPUP conversion MARGIN RELATIVE COLOR SCHEME 4
```

Conversion Type: Servoy App Menu JavaScript 2

Convert 3

Output Code:

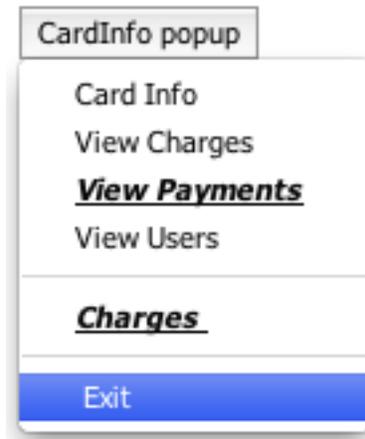
```
function menu1_example_convpad_apm()
{
  var appMenu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);
  var menu0 = appMenu.addMenu('Conversions', null, null, 'c');
  var submenu1 = menu0.addMenuItem('Area', null, null, 'e');
  var submenu2 = menu0.addMenuItem('Length', null, null, 'l');
  var submenu3 = menu0.addMenuItem('Mass', cmdblue_click , null, 's'); // _screen.activeform.otoolbar.cmdBlue.click
  var submenu4 = menu0.addMenuItem("", cmdwhite_click , null); // _screen.activeform.otoolbar.cmdWhite.click
}

function menu1_example_cardpad_apm()
{
  var appMenu = plugins.window.getMenu(plugins.window.getMenuCount() - 1);
  var menu0 = appMenu.addMenu('Card Info', null, null, 'i');
  var submenu1 = menu0.addMenuItem('View Charges', cmdred_click , null, 'v'); // _screen.activeform.otoolbar.cmdRed.click
  var submenu2 = menu0.addMenuItem('<html><b><i><u>View Payments</b></i></u></html>', cmdgreen_click , null, 'g');
  var submenu3 = menu0.addMenuItem('View Users', null, null, 'w');
  var submenu4 = menu0.addSeparator();
  var submenu5 = menu0.addMenuItem('<html><b><i><u>Charges </b></i></u></html>', null, null, 'c');
  var submenu6 = menu0.addSeparator();
  var submenu7 = menu0.addMenuItem('Exit ', null, 'media://Image1.bmp', 'x');
}
```

The development conversion tab is designed to provide a place where a single menu script can be quickly pasted and converted, making it easy to observe the differences between the App Menu and PopUp Menu JavaScript generated code output.

(1) Click the Clear Source Code button to clear the contents of the Visual FoxPro Menu Source Code field. Copy menu source code from any of your Visual FoxPro .MPR files, and paste into the Source Menu field. (2) Select the type of JavaScript menu code to generate, (3) click the Convert button. The code will be converted and the contents of the Output Code field will be copied and put onto the clipboard - ready for pasting into the Servoy Eclipse IDE.

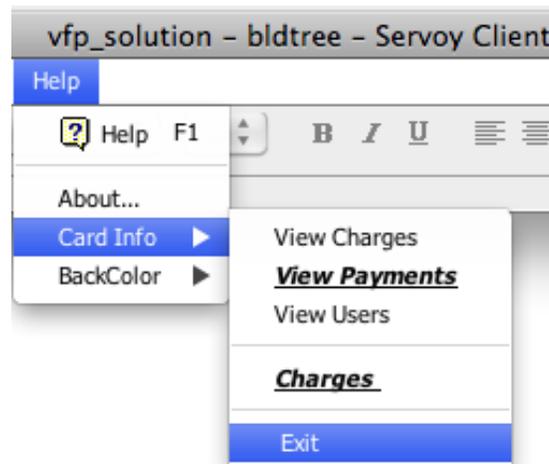
Servoy PopUp Menu JavaScript



Servoy popup menus are typically attached to form objects, and are useable within SmartClient and WebClient deployed applications. The generated JavaScript code automatically displays the menu above/below the clicked object. PopUp menus can also be displayed at specific coordinate locations on the form canvas, by modifying the generated code.

The generated JavaScript code for Servoy PopUp menus is written into a file named `menus_ppm.js` within the output directory for production conversions.

Servoy Application Menu JavaScript



A Servoy application menu can only be used within the SmartClient. Each menu appears under the last menu in the list of menu items. Menu items are displayed as submenu items.

The generated JavaScript code for Servoy Application menus is written into a file named `menus_apm.js` within the output directory for production conversions.

Visual FoxPro .PRG Code Conversion

Visual FoxPro Code Converter - Servoy

FmPro Migrator Developer Edition includes the MacOS X and Windows versions of the Visual FoxPro Code Converter as a compiled stand-alone application. This application converts FoxPro .PRG files and embedded form object code into Servoy JavaScript code.

Visual FoxPro Code Converter 1.17

1/12/2011

Introduction

The Visual FoxPro Code Converter included with FmPro Migrator Developer Edition reads the imported code from FoxPro .PRG files and converts the code into Servoy JavaScript code. Code can be converted individually, as shown using the Development tab, or can be converted for an entire Visual FoxPro project all at once.

Production Conversion Tab

Visual FoxPro Code Converter 1.17

www.FmProMigrator.com

This utility converts Visual FoxPro source code for use with other development environments.

Production Development

Output Directory: C:/ds/VFP_Tests/Solution_VFP_Fmig/ Browse (1)

Conversion Type: Servoy JavaScript

Convert (2)

Statistics:

6 Scripts Processed in 92 ms.

Total Lines Processed: 1426
% Lines Processed: 94%
Blank Lines Processed: 287
% Blank Lines Processed: 20%
Comment Lines Processed: 245
% Comment Lines Processed: 17%
Unsupported Lines Processed: 141
% Unsupported Lines Commented: 9%
UnProcessed Lines: 75
% UnProcessed Lines: 5%

----- UnProcessed Keywords List -----

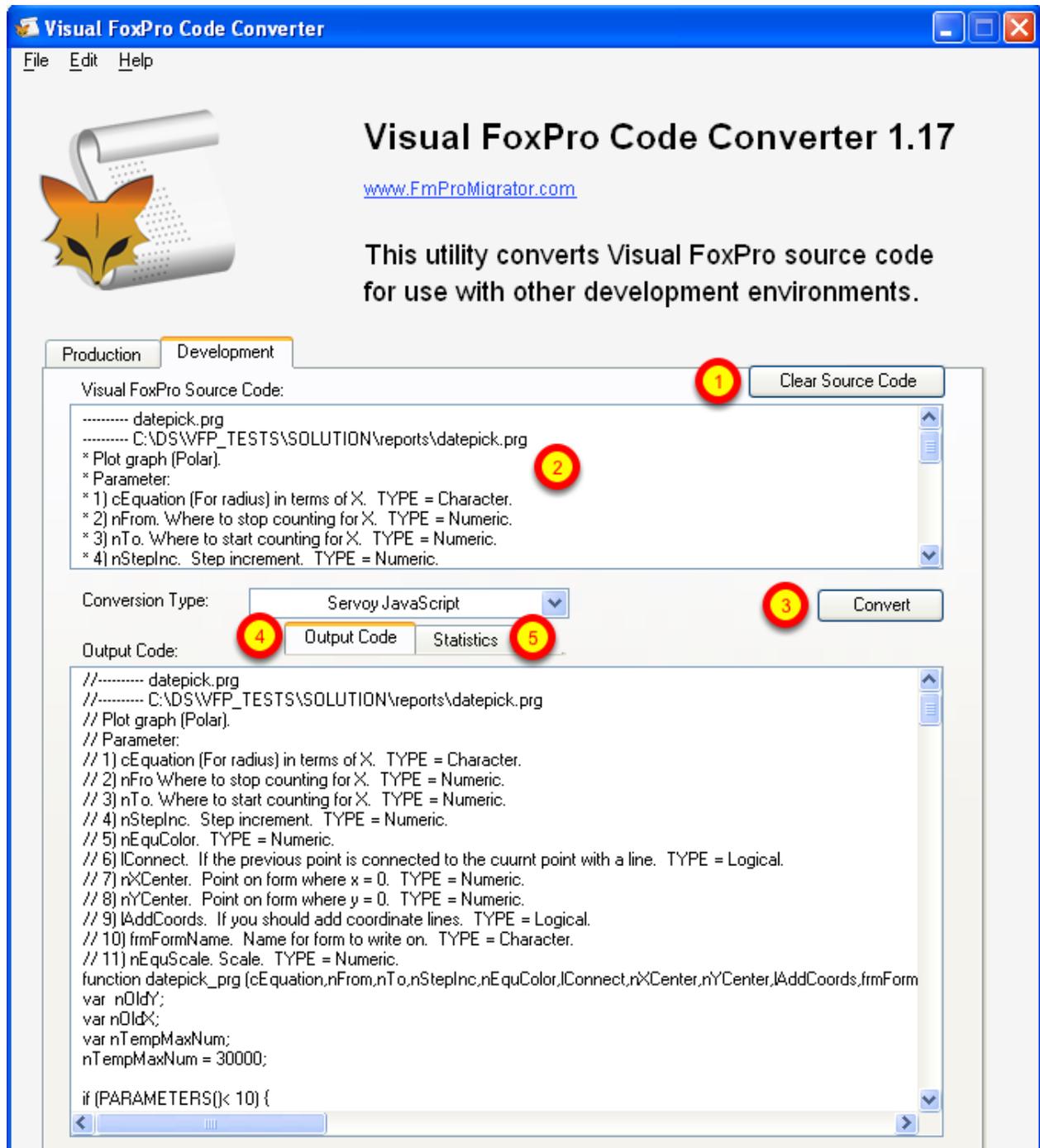
```
10 THISFORMSET.frmFD.DrawColor (  
9 frmFormName.line(nXCenTer,0,nXCenTer,frmFormName.height)  
9 frmFormName.line(0,nYCenter,frmFormName.width,nYCenter)  
5 frmFormName.line(nOldX+nXCenTer,nYCenter-nOldY,tempplotx+nXCenTer,nYCenter-tempploty)  
5 frmFormName.pset(tempplotx+nXCenTer,nYCenter-tempploty)  
4 frmFormName.pset(nXCenTer+(nXTemp*nEquScale),nYCenter-y)  
4 frmFormName.line(nXCenTer+(nOldX*nEquScale),nYCenter-nOldY,nXCenTer+(nXTemp*nEquScale),nYCenter-nY)  
2 thisforline(this.oldx,this.oldy,this.currentx,this.currenty)  
2 THISFORMSET.frmFD.SetCaption  
2 THISFORMSET.frmFD.ClearForm  
2 THISFORMSET.frmFD.ChgDrawMode  
2 THISFORMSET.frmFD.SetPenWidth (  
2 THISFORMSET.frmFD.GraphDemo (  
2 Main.Show  
2 THISFORMSET.frmFD.SetDrawMode (  

```

The Production tab will read and process all of the script records written into the FmPro Migrator MigrationProcess.db3 project database file during the Visual FoxPro conversion process. To perform a production conversion, (1) select the FmPro Migrator project folder containing the

MigrationProcess.db3 file, (2) click the Convert button. (3) The conversion statistics will be displayed within the Statistics field below the Convert button. The generated Servoy JavaScript files will be written into a folder named VFP Converted Scripts within the output folder.

Development Conversion Tab

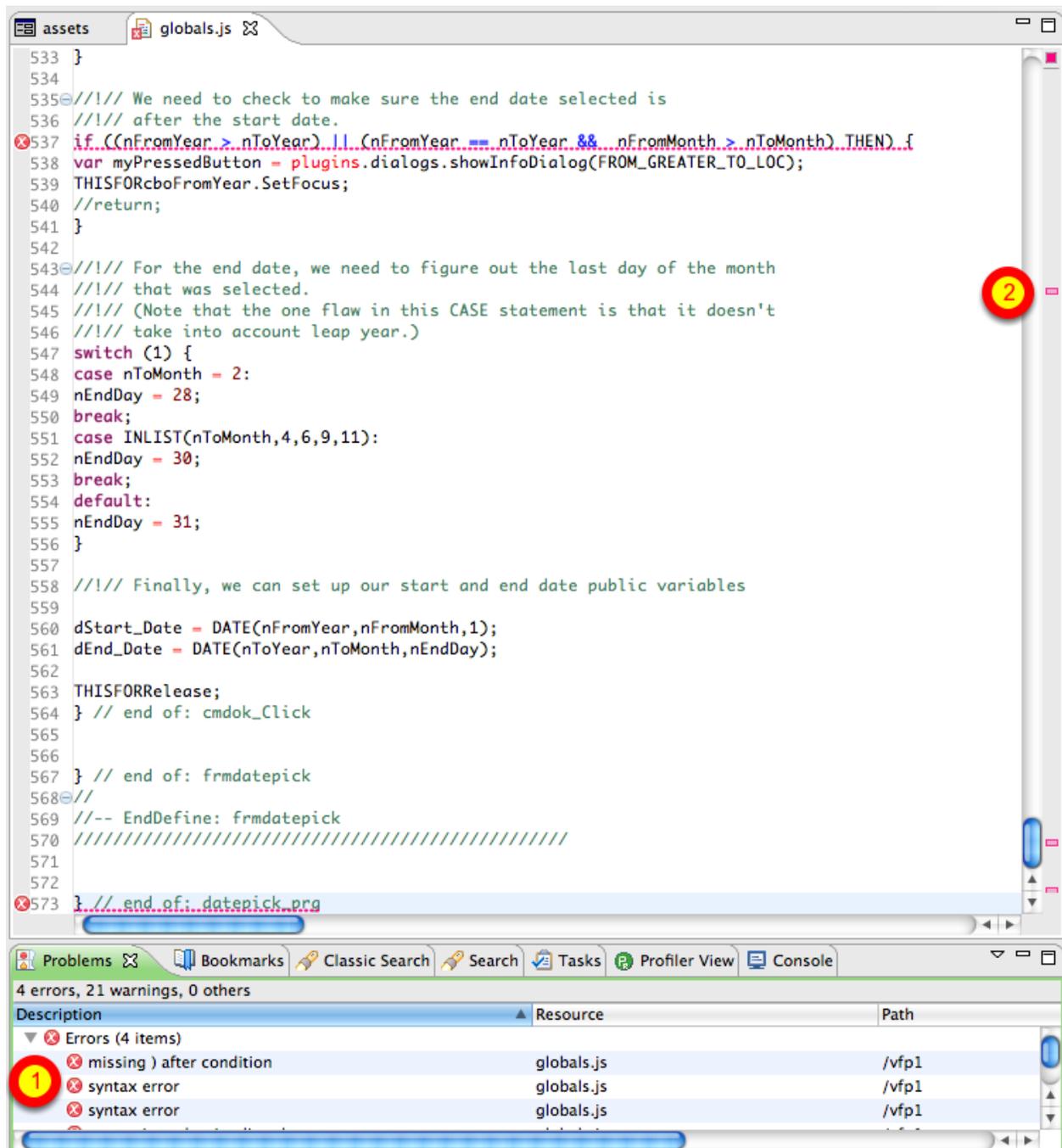


The development conversion tab is designed to provide a place where a single FoxPro script can be quickly pasted and converted.

(1) Click the Clear Source Code button to clear the contents of the Visual FoxPro Source Code field.

Copy source code from any of your Visual FoxPro .PRG files, and (2) paste the code into the Source Code field. (3) click the Convert button. The code will be converted and the contents of the Output Code field will be written into the Output Code field and will also be put onto the clipboard - ready for pasting into the Servoy Eclipse IDE. (5) Clicking the Statistics tab will display the conversion statistics for the converted script.

Pasting Converted Code into the Servoy Eclipse IDE

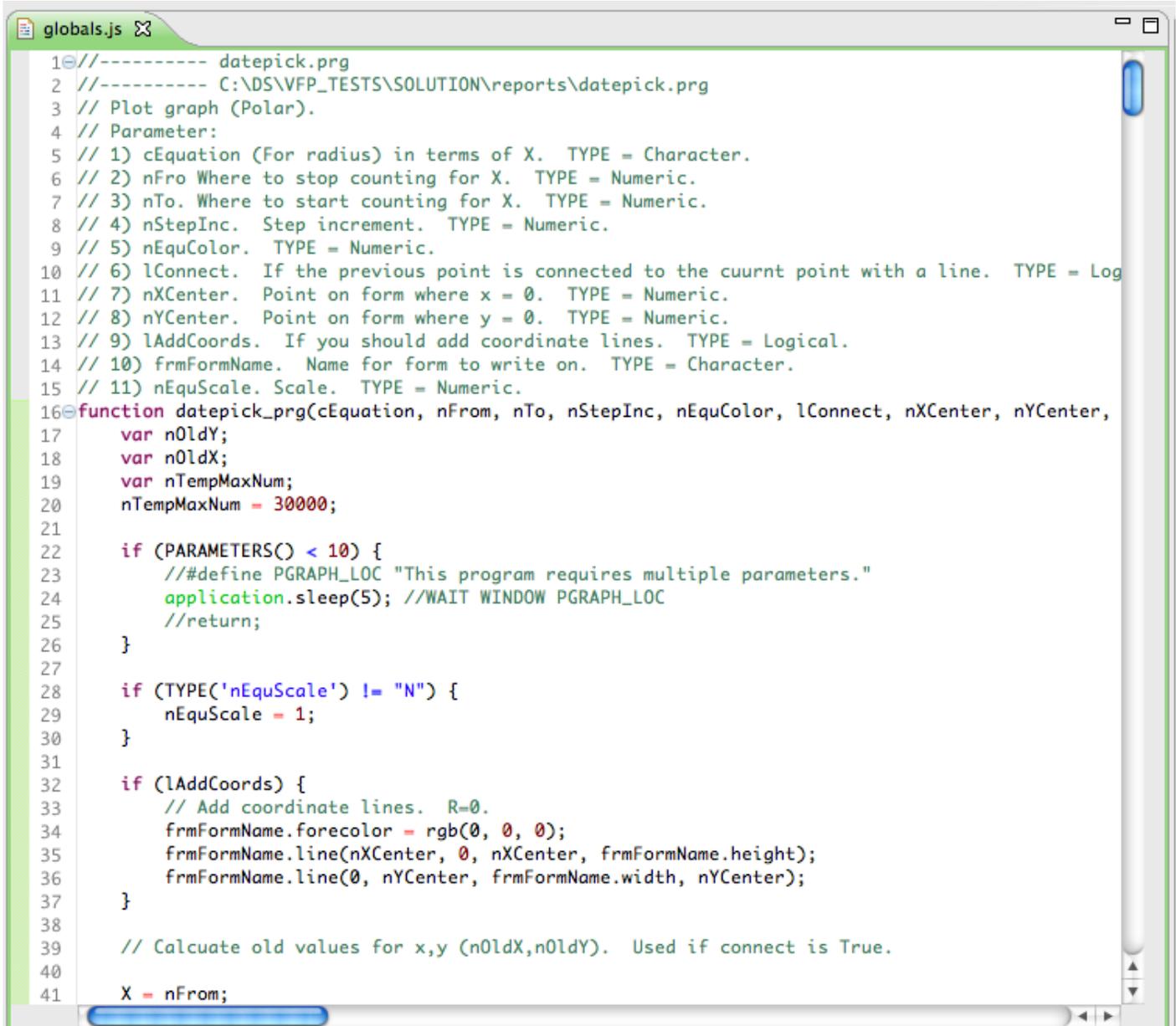


Converted files can be copied and pasted directly into the Servoy Eclipse IDE, or the JavaScript files can be moved into the project folder for editing. However, this is not necessarily the best way to work with converted code files due to the structure of the original Visual FoxPro code. Generally, it is

recommended that the code be pasted and modified in small blocks.

This screenshot shows a 573 line script named datepick.PRG which was converted and pasted into a new globals.js file within the Servoy project. (1) Note that the converted script is showing 4 errors within the Problems tab below the script editor window. (2) Also, the red error bars are displayed along the right hand edge of the editor window. These 4 errors represent a 0.7% error rate for the converted lines of code. Typically the converted script error rate will be 0% - 3%, and these errors will at least need to be commented out in order for the Servoy Eclipse IDE to format and parse the code.

Commenting Errors & Formatting Converted Code



```
1 //----- datepick.prg
2 //----- C:\DS\VFP_TESTS\SOLUTION\reports\datepick.prg
3 // Plot graph (Polar).
4 // Parameter:
5 // 1) cEquation (For radius) in terms of X. TYPE = Character.
6 // 2) nFrom Where to stop counting for X. TYPE = Numeric.
7 // 3) nTo. Where to start counting for X. TYPE = Numeric.
8 // 4) nStepInc. Step increment. TYPE = Numeric.
9 // 5) nEquColor. TYPE = Numeric.
10 // 6) lConnect. If the previous point is connected to the current point with a line. TYPE = Logical.
11 // 7) nXCenter. Point on form where x = 0. TYPE = Numeric.
12 // 8) nYCenter. Point on form where y = 0. TYPE = Numeric.
13 // 9) lAddCoords. If you should add coordinate lines. TYPE = Logical.
14 // 10) frmFormName. Name for form to write on. TYPE = Character.
15 // 11) nEquScale. Scale. TYPE = Numeric.
16 function datepick_prg(cEquation, nFrom, nTo, nStepInc, nEquColor, lConnect, nXCenter, nYCenter,
17     var nOldY;
18     var nOldX;
19     var nTempMaxNum;
20     nTempMaxNum = 30000;
21
22     if (PARAMETERS() < 10) {
23         //#define PGRAPH_LOC "This program requires multiple parameters."
24         application.sleep(5); //WAIT WINDOW PGRAPH_LOC
25         //return;
26     }
27
28     if (TYPE('nEquScale') != "N") {
29         nEquScale = 1;
30     }
31
32     if (lAddCoords) {
33         // Add coordinate lines. R=0.
34         frmFormName.forecolor = rgb(0, 0, 0);
35         frmFormName.line(nXCenter, 0, nXCenter, frmFormName.height);
36         frmFormName.line(0, nYCenter, frmFormName.width, nYCenter);
37     }
38
39     // Calculate old values for x,y (nOldX,nOldY). Used if connect is True.
40
41     X = nFrom;
```

Making a few changes to the code results in resolving the 4 errors. Selecting Source -> Format formats the code and applies proper indenting to make the code easier to read. The code can now be tested with the debugger or edited further.

In general, comments should be located within the method they reference, instead of being above the function definition as is commonly done within FoxPro code - or the comments may be removed during the code formatting process. The converted code has also been changed a little bit from the original in an attempt to wrap the contents of .PRG code within a function, by adding a function definition in place of the first PARAMETER statement near the top of the code. A closing curly bracket was also added at the end of the code as well.

Understanding the Conversion Statistics

The code conversion statistics are designed to provide a general idea of the percentages of processed, unprocessed and unsupported lines of code which were converted. Here is an explanation for these statistics:

Total Lines Processed: The total number of lines which were processed.

% Lines Processed: - The percentage of lines for which some type of processing was done. Each line will either be modified or passed thru unchanged. The following statistics lines provide more details of what was done during the processing.

Blank Lines # or %: - Blank lines will be counted as processed, because nothing needs to be done to them even during a manual conversion since white spaces aids code readability.

Comment Lines # or %: - Comment lines are modified by changing the FoxPro comment character with the JavaScript comment character.

Unsupported Lines # or %: Unsupported Lines are counted in the processed lines percentage, because the lines have been analyzed and determined to be unusable within the context of a multi-user server based environment. A detailed listing of the Unsupported Keywords is provided at the end of the report. These lines are commented to prevent errors.

UnProcessed Lines # or %: The UnProcessed lines are not counted within the total processed percentage, because they did not match an implemented keyword. Each line is passed thru with a semicolon added to the end of the line. Lines containing the equality operator are not considered UnProcessed, and they are also passed thru with a semicolon added to the end of each line.

UnProcessed Keywords List - During script processing, a log is kept of the keywords which were not processed because they did not match a known FoxPro keyword. The number of times each keyword was found is logged along with the first two words of the instruction line. This list provides a quick way to determine if there are FoxPro keywords which need to be added to the conversion process.

Unsupported Keywords List - Each Unsupported Keyword is logged during processing along with a number indicating how many times it was found during processing. This list can provide a quick way to review whether there are FoxPro keywords which could potentially be added to the list of supported keywords in the future. For instance if a particular keyword is observed many times, and a logical replacement can be for the keyword, then the process can be improved considerably.

preParse Processing

Prior to the line-by line processing of each line of source code, a preParse processing step is performed on each script.

The tasks performed at this stage of processing include:

Removal of Line Continuation Characters - The Visual FoxPro line continuation character is a semicolon, which conflicts with the JavaScript line ending character - therefore it must be removed for compatibility purposes. Also, concatenating each of the individual lines together into one single new line makes it easier to parse the original line of code. The total number of lines counted for statistical purposes represents the number of lines which exist in the source script after removing the line continuation characters.

Common FoxPro Function Replacement - Some commonly used FoxPro functions, operators and constants have direct replacements within Servoy JavaScript code. The following replacements are made across the entire script:

Operator and Constant Replacement

AND => &&

OR = ||

NOT = !

<> => !=

m. => empty

.T. => true

.F. => false

tab => empty

Function Replacement

UPPER()=> toUpperCase()

LOWER() => toLowerCase()

LEN() => str.length()

VAL() => parseFloat()

INT() => parseInt()

LTRIM() => stringTrim()

RTRIM() => stringTrim()

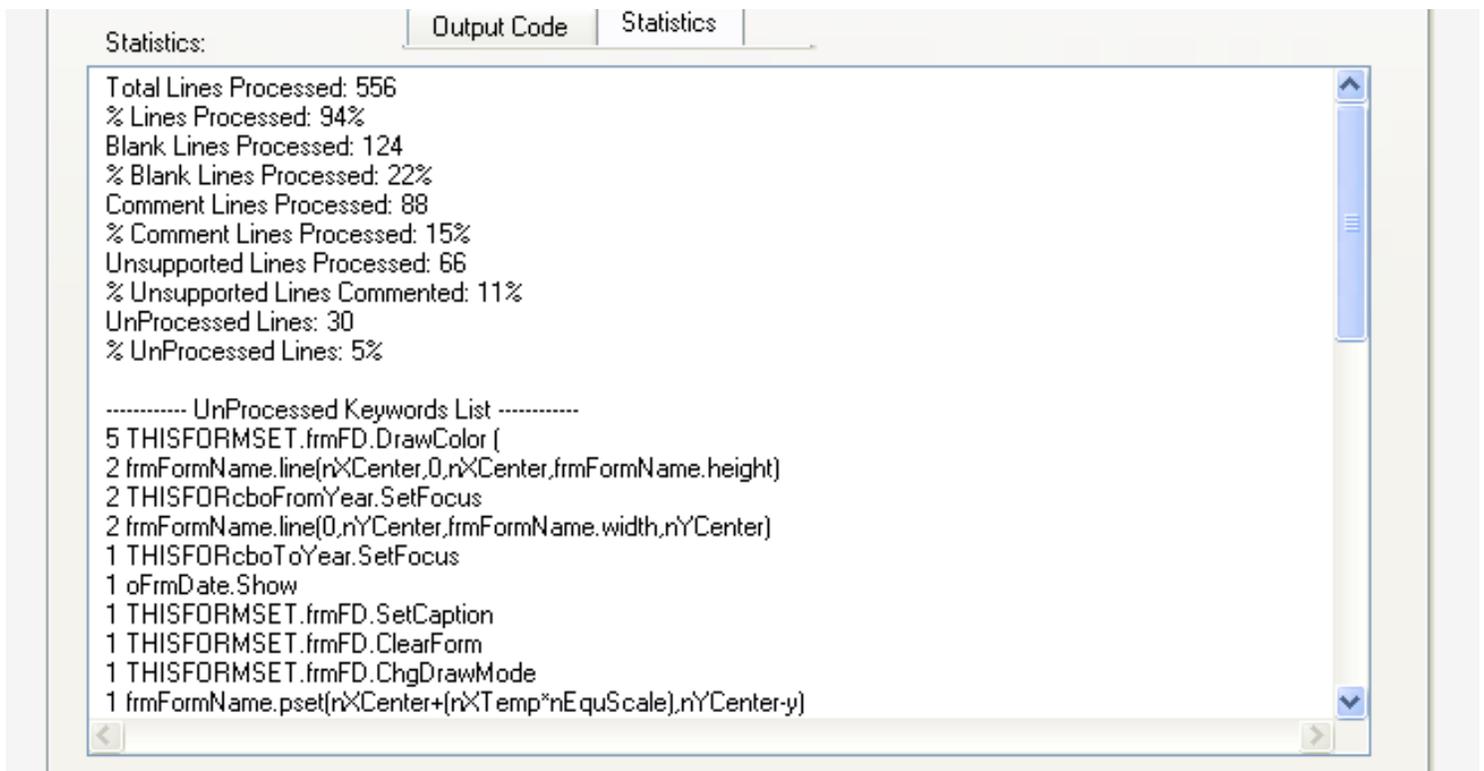
ALLTRIM() => stringTrim()

SUBSTR() => str.substr()

CHR() => String.fromCharCode()

ASC() => charCodeAt()

Manual Code Changes



```
Statistics:
Output Code
Statistics

Total Lines Processed: 556
% Lines Processed: 94%
Blank Lines Processed: 124
% Blank Lines Processed: 22%
Comment Lines Processed: 88
% Comment Lines Processed: 15%
Unsupported Lines Processed: 66
% Unsupported Lines Commented: 11%
UnProcessed Lines: 30
% UnProcessed Lines: 5%

----- UnProcessed Keywords List -----
5 THISFORMSET.frmFD.DrawColor (
2 frmFormName.line(nXCenTer,0,nXCenTer,frmFormName.height)
2 THISFORMcboFromYear.SetFocus
2 frmFormName.line(0,nYCenter,frmFormName.width,nYCenter)
1 THISFORMcboToYear.SetFocus
1 oFrmDate.Show
1 THISFORMSET.frmFD.SetCaption
1 THISFORMSET.frmFD.ClearForm
1 THISFORMSET.frmFD.ChgDrawMode
1 frmFormName.pset(nXCenTer+(nXTemp*nEquScale),nYCenter-y)
```

The automated conversion processing done by the Visual FoxPro Code Converter will attempt to make as many changes as is practical to convert FoxPro code into compatible Servoy JavaScript code. There are however a variety of changes which can only be performed manually.

Code restructuring - A FoxPro PRG file can contain bare keywords outside of a Function/Procedure because the whole file is basically treated like a procedure. Servoy developers generally write all code encapsulated within a method (or function). Also, most Servoy methods don't define additional methods, so these will generally need to be restructured to become separate methods.

Windows Client Based Coding - Commands which attempt to write directly to a DBF file on a user's local disk or open/close database files are unsupported, and unneeded within a Servoy application. All database connections are available immediately just by specifying the dataProvider. And all tables are stored centrally instead of on the user's local hard disk.

Visual FoxPro Specific Coding - The SET, SYS() and FORMSET commands are very specific to functionality which only exists within FoxPro - therefore these commands are commented out and marked as unsupported. FoxPro specific functions also come into this category, though some commonly used functions are converted during the preParsing part of the conversion process.

Form Object Creation Coding - Servoy forms are built by the developer (or converted by FmPro Migrator directly from the original FoxPro forms/reports/labels). Generally, Servoy objects are not created on forms programmatically by the developer, so the FoxPro ADD OBJECT commands are

commented out as unsupported.

Command Line Coding - FoxPro projects may contain commands which prompt a user to enter an answer in a command window and wait for the user to enter an the answer. Processing continues once the user has entered the answer followed by a return key. Servoy apps are event driven apps designed with a graphical interface instead of being command line driven apps. In a Servoy app, the user enters info into fields on a form and then clicks in the form background to commit the record or clicks a button to initiate the processing. Therefore the processing of the entered info could be done via a calculated field or by the button click event.

KeyBoard Event Driven Coding - FoxPro code may contain keystroke driven event code. A Servoy application cannot be dependent upon trapping a particular keystroke combination in order to trigger the execution of code because the application may be deployed via a web browser.

List of Supported Keywords

There are 60 keywords supported during the code conversion process. These keywords are:

*

&&

+

ACTI WIND

ACTIVATE WINDOW

ALLTRIM()

AND

CASE

CASE RETURN

CHR()

CHR()

COUNT

CREATE TABLE

DIMENSION

DISPLAY DATABASE

DISPLAY TABLES

DO

DO CASE

DO FORM

DO WHILE

ELSE

ENDCASE

ENDDEFINE

ENDDO
ENDFOR
ENDFUNC
ENDIF
ENDIF
ENDPROC
.F.
FOR
FOUND()
FUNCTION
IF
INSERT INTO
INT()
LEN()
LOCAL
LOOP
LOWER()
LTRIM()
MESSAGEBOX
MODIFY DATABASE
NOT
OR
OTHERWISE
PRIVATE
PROCEDURE
PUBLIC
QUIT
RETURN
RTRIM()
SHOW WINDOW
SORT
STORE
SUBSTR()
.T.
UPPER()
VAL()
WAIT

List of UnSupported Keywords

There are 55 keywords which are unsupported during the code conversion process. These keywords are:

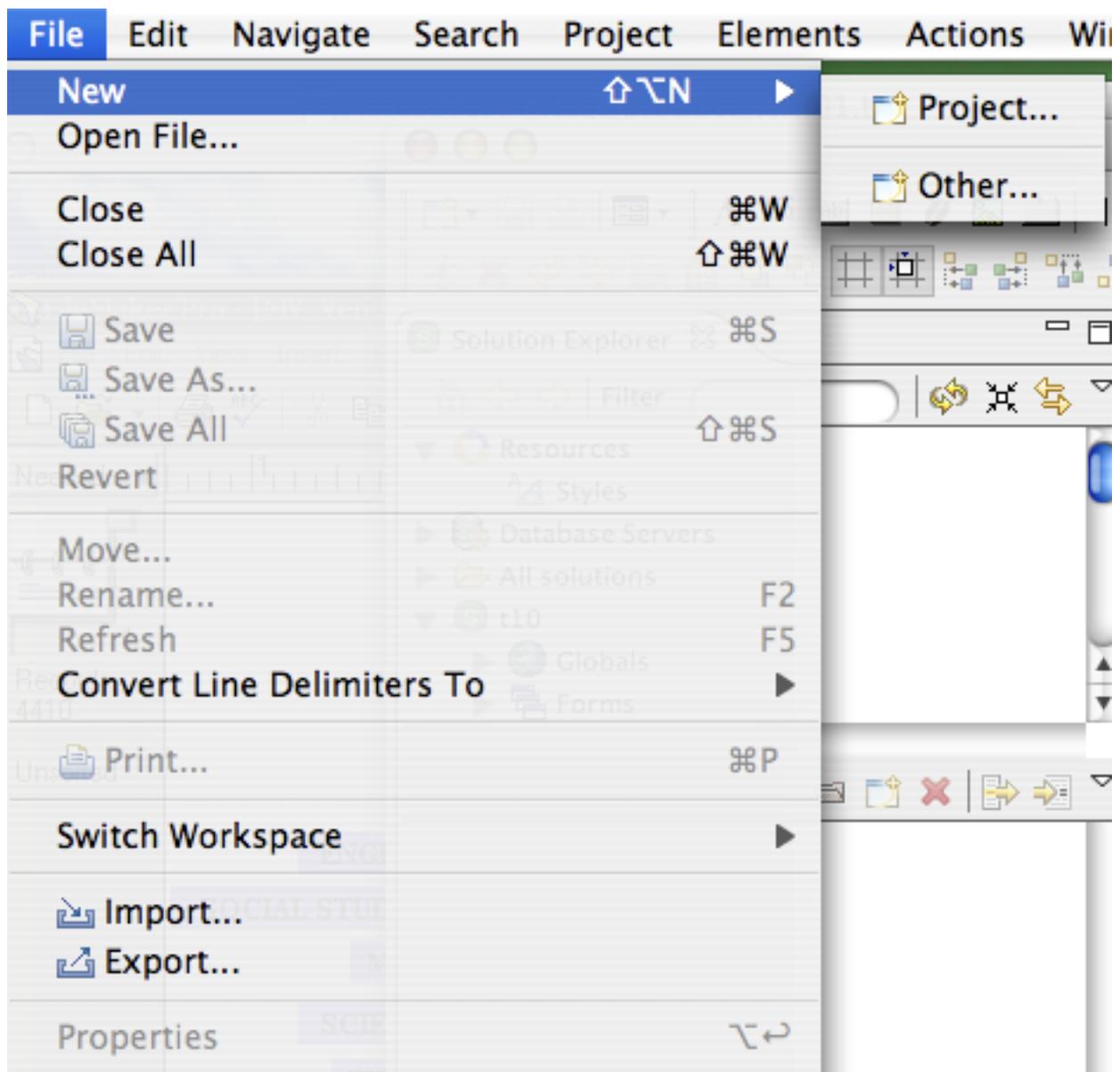
\$
?
ADD OBJECT
APPEND
BOF()
BROWSE
CLEAR ALL
CLEAR EVENTS
CLOSE ALL
CLOSE DATA
CLOSE DATABASE
CLOSE TABLES
CONTINUE
CREATE
CREATE DATABASE
CTOD()
DEACTIVATE WINDOW
#DEFINE
DEFINE WINDOW
DELETE
DELETED()
DISPLAY
DISPLAY
DTCO()
EOF()
GO
GOTO
HIDE WINDOW
#INCLUDE
INDEX
LIST
LOCATE
MODIFY STRUCTURE
ON ERROR
ON KEY
OPEN DATABASE

PACK
POP KEY
PUSH KEY
READ
RECALL
RECNO()
RELEASE
RELEASE ALL
RELEASE CLASSLIB
RELEASE THISFORMSET
RELEASE WINDOW
REPLACE
SELECT
SET
SKIP
SUM
THISFORMSET
USE
ZAP

Servoy Conversion

Create Servoy Eclipse Project

Create New Servoy Eclipse Project



Select New -> Project from the File menu to create a new Servoy Eclipse 4/5 project.

This step is necessary, because FmPro Migrator will delete and re-create the directory when creating the new Servoy Eclipse project from the FileMaker Pro database structure information stored within the FmPro Migrator MigrationProcess.db3 file.

Enter project name

New solution

Create a new solution
- a new Servoy solution project will be created

Solution name **t2**

Solution type Normal

Please choose the resources project this solution will reference (for styles, column/

Create new resource project resources

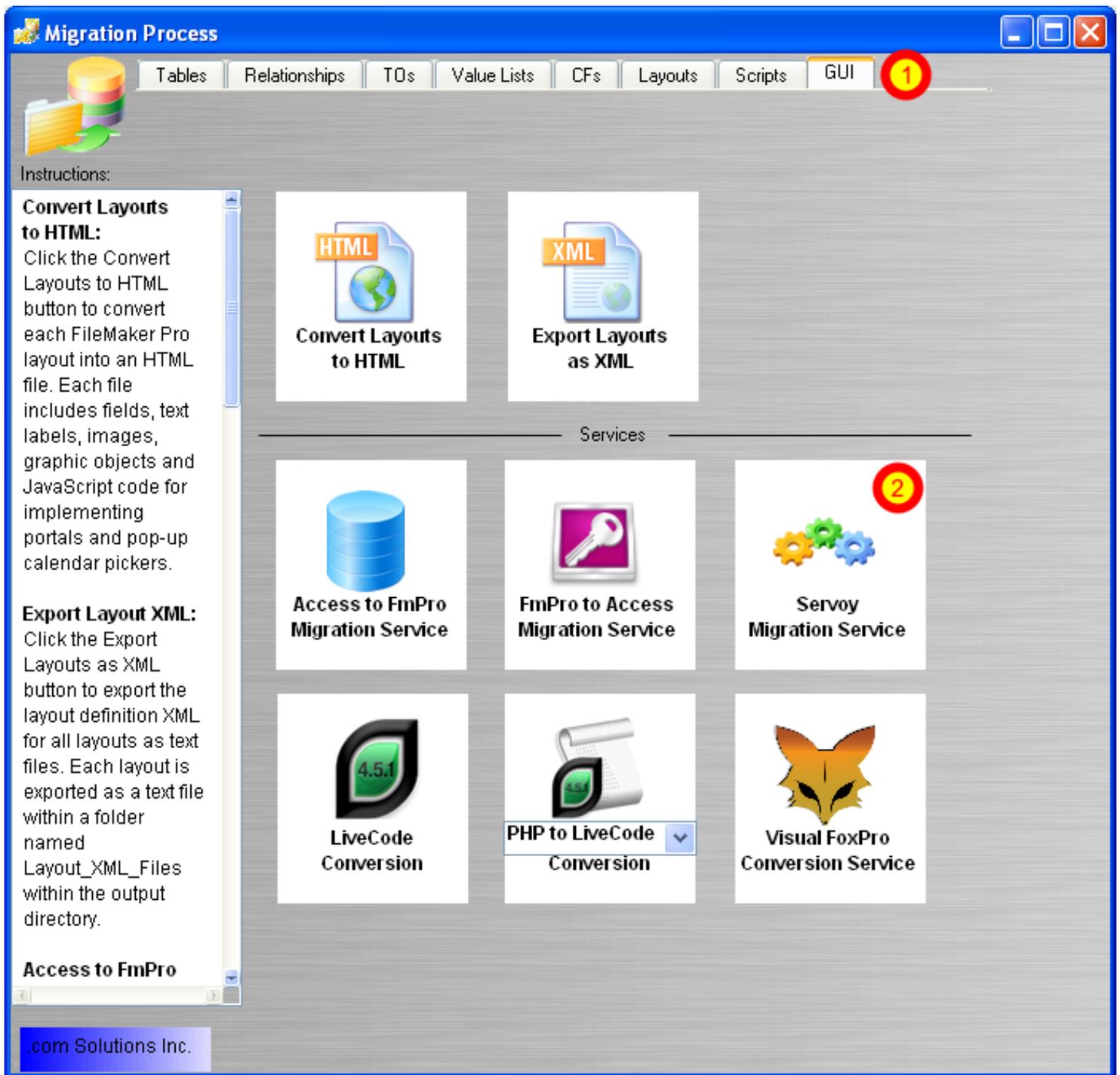
Use existing resource project resources

? < Back Next > Cancel **Finish**

Enter the project name, then click the Finish button.

Enter Servoy Project Parameters

Click FmPro to Servoy Migration Service Button



The screenshot shows the 'Migration Process' window with the 'GUI' tab selected. A red circle with the number '1' highlights the 'GUI' tab. Below the tabs, there are instructions for 'Convert Layouts to HTML' and 'Export Layout XML'. A 'Services' section contains several buttons: 'Access to FmPro Migration Service', 'FmPro to Access Migration Service', 'Servoy Migration Service' (highlighted with a red circle and the number '2'), 'LiveCode Conversion', 'PHP to LiveCode Conversion', and 'Visual FoxPro Conversion Service'. The bottom of the window shows the text '.com Solutions Inc.'

- 1) Click on the GUI folder tab within the FmPro Migrator Migration Process window.
- 2) Click on the Servoy Migration Service button.

Servoy Project Parameter Fields - Demo Mode

Instructions:

Migrate FileMaker Pro Databases to Servoy:

Step 1) Import information about the FileMaker Pro database(s) into FmPro Migrator Developer Edition following the Table Consolidation Procedure PDF.

Step 2) Create a new Servoy Eclipse 4.1+ project. Enter the new Project Name, Database Connection name and select the new project workspace directory.

Layout Qty: 87

Project Name: vfp1_solution

DB Connection: mysql_mbp17

Project Directory: C:/Documents and ... Browse

Processing Type: Demo Order License Key

QUOTE

Migrate

By default, FmPro Migrator sets the Processing Type for the FmPro to Servoy Migration Service to Demo mode. When running in Demo mode, 5 layouts and 5 scripts are processed - and all other objects including relationships, globals and value lists are migrated.

(1) Enter the Servoy project name which was just created, (2) the Servoy database connection name, (3) and select the project directory within the Servoy project workspace.

Note: Since FmPro Migrator deletes and re-creates the project directory each time the Migrate button is pressed, there are a couple of safety features incorporated into the Project Directory Browse button. The selected output directory path must contain the text "workspace" and it must contain the text consisting of the Project Name ("t2" in this example).

To enter the Servoy Migration license key, select "Licensed" from the Processing Type menu, copy the license key to the clipboard from your email program, click on the clipboard icon to the left of the license key field. The license key will be copied from the clipboard and validated.

Servoy Migration - Migrate Button

Click Migrate Button

Instructions:

Migrate FileMaker Pro Databases to Servoy:
Step 1) Import information about the FileMaker Pro database(s) into FmPro Migrator Developer Edition following the Table Consolidation Procedure PDF.

Step 2) Create a new Servoy Eclipse 4.1+ project. Enter the new Project Name, Database Connection name and select the new project workspace directory.

Layout Qty: 87

Project Name: vfp1_solution

DB Connection: mysql_mbp17

Project Directory: C:/Documents and Settings/... Browse

Processing Type: Licensed

License Key: [redacted]

Valid License Key
Valid Until 11/12/2011 (87 Forms/Reports)

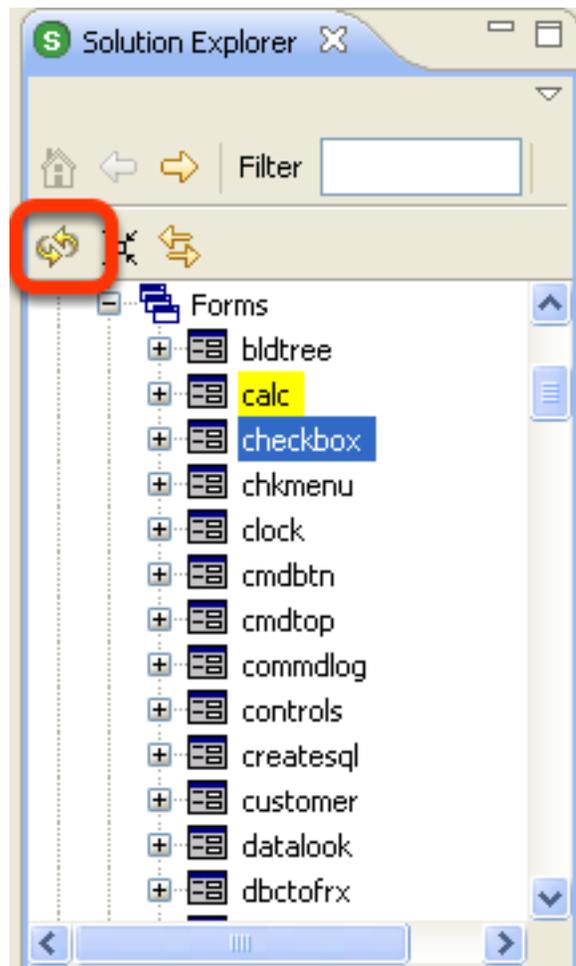
87 Layouts Processed in 10.8 Sec.
131 Scripts Processed.

Migrate

Clicking the Migrate button starts the conversion process for the Layouts, Value Lists and Scripts stored within FmPro Migrator. The selected Servoy project is deleted and completely re-created during the migration process.

Servoy Migration - Refresh Project

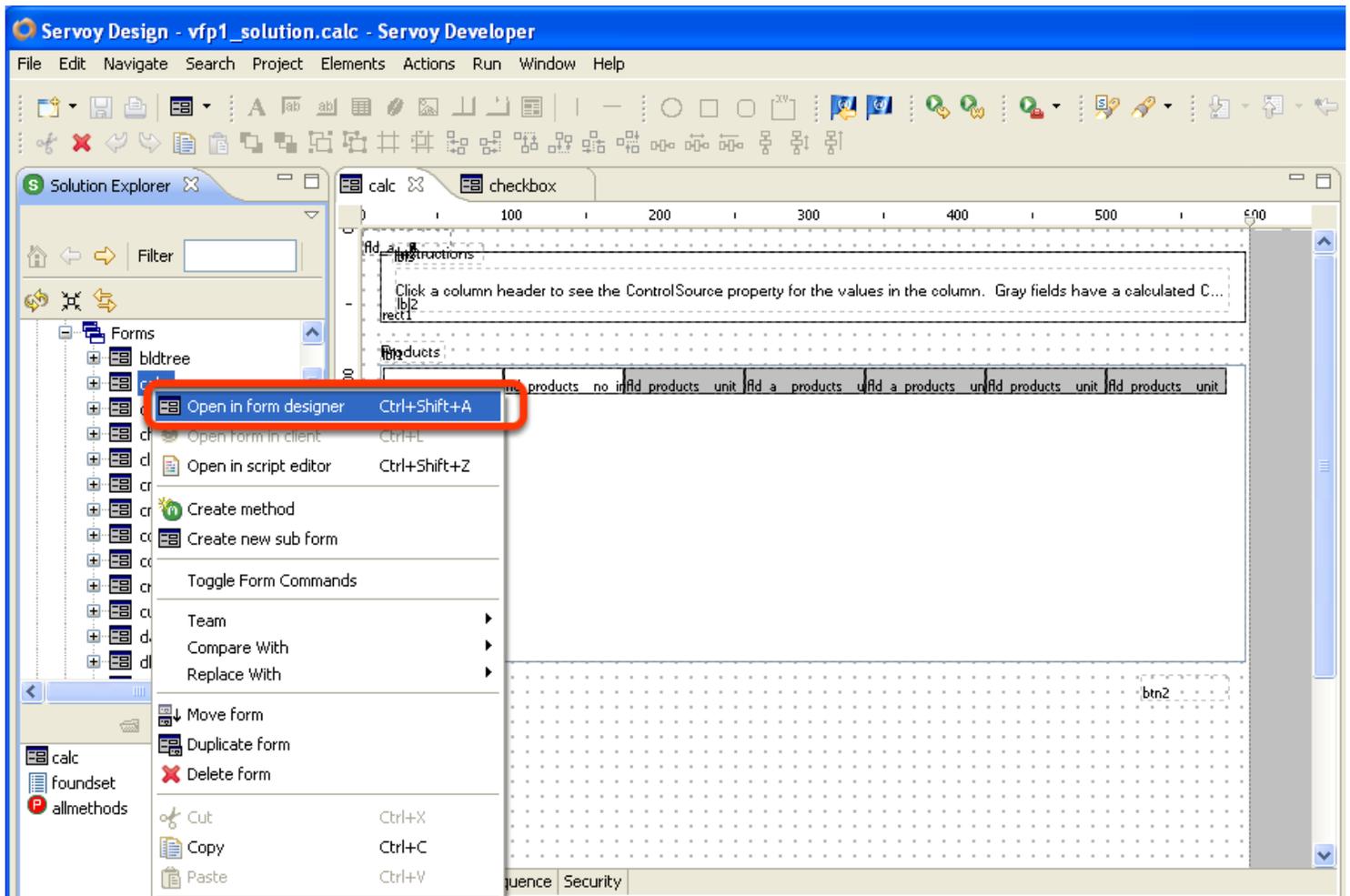
Refresh Servoy Project



Click the Refresh View button in the Servoy Eclipse IDE after generating the new project. The Servoy Eclipse IDE may prompt you to update the project to the latest file format version.

Note: Currently, Servoy projects created by FmPro Migrator are created using the Servoy 4.1 file format. New versions of the Servoy Eclipse IDE will automatically upgrade the file format to version 5.

Open & View Converted Forms



Right click on any form and select the Open in form designer menu item to view the converted form in the Servoy Eclipse IDE.

Post Migration Tasks - Visual FoxPro to Servoy Migration

Some manual steps will be required after the automated processing has been completed.

Unsupported Form/Report Features

There are some elements incorporated within Visual FoxPro Forms/Reports which are not currently supported Servoy Migration Service. These elements include report parts (Header/Footer, Sub summary sections) and ActiveX controls embedded on forms.

Layout Parts - The Layout XML format does not include information about layout parts, so this information cannot be processed automatically. However FmPro Migrator does automatically create a body part for each Servoy Form which is created, in order to insure that you can immediately see the objects within each form. It is easy to add/remove/change form parts within the Servoy Eclipse IDE.

Object Tab Order - The Layout XML format does not include information defining the tab order of each object on the layout, so this information cannot be passed thru from the original project file.

Object Z-order Stacking - The Layout XML format does not include information about the front to back stacking order of the objects on the layout, so this information cannot be passed thru from the original project file. However FmPro Migrator does assign a formIndex parameter for each form object in order to reduce the possibility of important objects like fields and text labels from getting covered up by objects like rectangles and images. Some manual changes may still be required after the automated conversion.

ActiveX Controls - It does not currently appear to be possible to read enough information about individual ActiveX controls on Visual FoxPro forms to determine what type of control or parameters are being implemented. Therefore FmPro Migrator simply creates a red box representing the object on the form. This way, FoxPro developers can quickly determine if there were any ActiveX controls on the converted form.

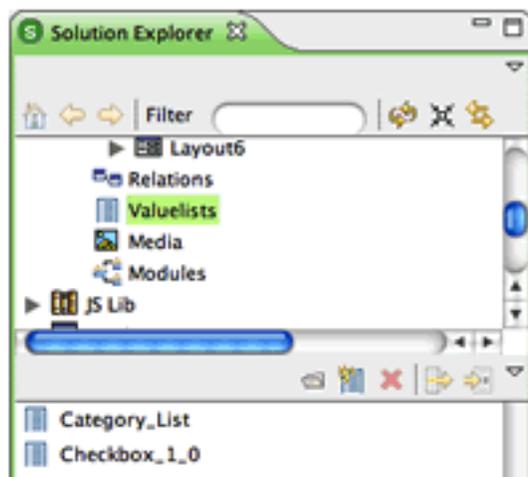
Many commonly used ActiveX controls have equivalent Java bean implementations which are either included or can be added to a Servoy Eclipse project. Some of these Java beans include beans which implement a TreeView or charting. Select the Place Bean... button from the toolbar at the top of the screen to display a list of installed Java beans. [See screenshot above.]

There are 3rd party Java beans available from the [Servoy Marketplace](#), as well as the [servoycomponents.com](#) website. Some of the Java beans and plug-ins included on the servoycomponents.com website include: Calendar Bean, Gantt Bean, Barcode and plug-in Bean, DDE Plug-in, HTMLeditor Plug-in, Cryptor Plug-in, Dialog Plug-in and FAX Plug-in.

Visual FoxPro Code to JavaScript Conversion

Every Visual FoxPro script is converted into lines of commented Javascript code, which is written into the Globals.txt file within the Servoy project directory. There is an easier to read uncommented version of the original scripts stored in the VFP_Scripts.txt plain text file within the FmPro Migrator Output Directory. This code is provided as a guide to the Servoy developer during the development of replacement Javascript code to replace the original business logic. It is important to note that a significant amount of Visual FoxPro code which was used for managing record locking and multi-user DBF file access will no longer be needed in the converted solution because the Servoy application server handles these tasks automatically.

Custom and Dynamic Value Lists



FmPro Migrator converts custom value lists into Servoy Value lists. However some custom value list names may have been duplicated within the original FileMaker solution if it was a multi-file solution. Duplicated Value Lists having the same name and contents will be skipped automatically by FmPro Migrator. Duplicated value lists having the same name but different contents will be renamed and added to the Duplicate Objects Report.xls file generated by FmPro Migrator. Manual work will be required if a renamed Value List contains information which is actually preferred over the first occurrence of the Value list which was created.

Dynamic value lists will require manual work to add them to Servoy and then associate the value list with the appropriate field objects.