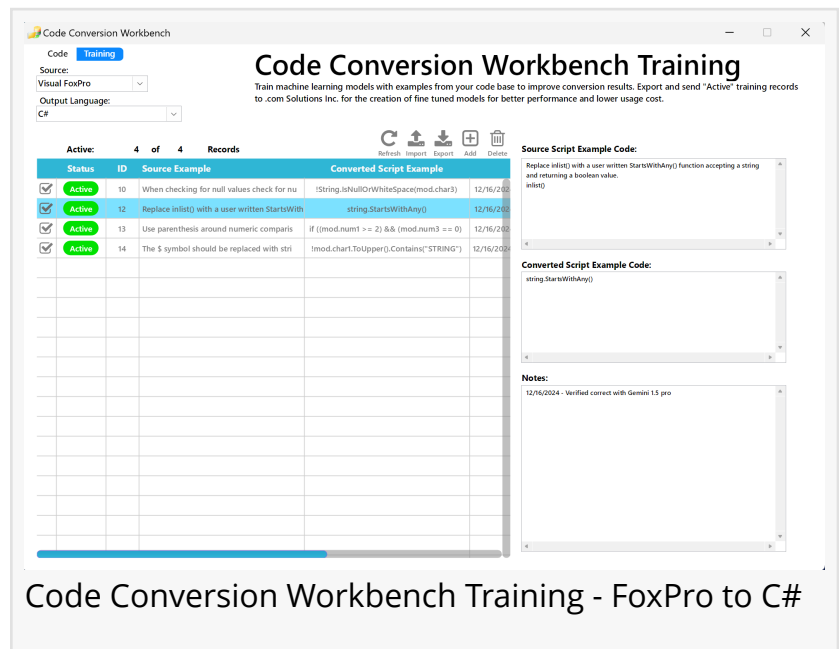# FmPro Migrator 11.28 Enhances Visual FoxPro to C# Conversion with Code Conversion Workbench AI Model Training Feature

Using simple english examples in the Code Conversion Workbench AI Model Training tab provides fine-grained control over how C# code is generated by the tool.

FREMONT, CA, UNITED STATES, January 21, 2025 /EINPresswire.com/ -- .com Solutions Inc. announces the release of FmPro Migrator AI Accelerated Edition 11.28. This release enhances Visual FoxPro to C# code conversions with a new Code Conversion Workbench Training feature. The training feature provides developers with fine-grained control over how LLMs generate code during the automated code conversion process.

Code Conversion Workbench Training - FoxPro to C#

Each FoxPro to C# project has unique needs in terms of the conversion process. Training LLMs customizes the generated code to match the needs of individual projects. For instance, some projects might need unique methods applied for text processing and comparison purposes. Other projects could benefit from using the VFP Toolkit for .NET. Each of these scenarios can be met by creating individual training records in the Training tab of the Code Conversion Workbench.

"

The Code Conversion Workbench AI Model Training feature of FmPro Migrator generates Visual FoxPro to C# converted code ready to compile in Visual Studio."

David Simpson

Using model training improves the conversion quality of lower-cost open-source models running through Ollama to closely match the output of higher-cost public models. This feature can be useful for situations in which LLMs need to be run locally in high-security computing environments without access to the internet.

How it works:

For each training record, an english text description is written to explain how the LLM should convert the FoxPro code into C# (or any other). This english text is the most important part of the instruction provided to the AI model.

The following examples are showing FoxPro to C# conversion, but this LLM training feature works across all programming language combinations.

Example #1 - FoxPro inlist()
In order to improve readability and maintainability in the C# code, we want to replace the VFP inlist() function with a user-written StartsWithAny() class.

Without training, the LLM creates individual comparisons for each option being checked.

Adding the training record results in the StartsWithAny() class being used within the code. As a bonus - the Google Gemini 1.5-Pro model has written the class for us and implemented method overloading to handle both string and integer comparisons.

Example #2 - FoxPro empty()
Instead of just checking for Null or Empty values, we also want to check for whitespace. By default, only Null or Empty values are checked.

But by adding the training record, the LLM will also check for whitespace in every place where it finds empty() in the VFP code base.

The result is that string.IsNullOrEmpty() will get replaced with String.IsNullOrWhiteSpace() in the generated code.

C# Converted Script With Training

```
public class Validator
{
public int ValidateOptions_299(Module mod, Job job)
{
if (mod.option == 249 && mod.num2 > 96 &&
StartsWithAny(job.option, "ABC", "DEF", "XYZ"))
{
string lcmsg = "This option is not available with a value over 8 feet.";
return -1;
}
return 0; // Or whatever the default successful return value should be
}

public static class StringExtensions
{
public static bool StartsWithAny(this string str, params string[] values)
{
return values.Any(str.StartsWith);
}

public static bool StartsWithAny(this int num, params int[] values)
{
return values.Contains(num);
}
}
}
```

Code Conversion Workbench - VFP inlist()
Example Code Generated - Hilited

FmPro Migrator Site License Edition Icon 256 x 256

Example #3 - FoxPro $

When converting the VFP $ operator, we want to use Contains, which is used by default with most public models. But when using Ollama to run the llama3 (8B) locally, we get the output shown on the website training examples. Therefore, using the training record enables the output of the local LLM to match the output of the much more expensive public LLMs.

Also, notice that the IsNullOrWhiteSpace() results shown in the above examples weren't generated. This is because that training record wasn't enabled for this row of results. These training records can and should be enabled at once in order to achieve the desired output.

Example #4 -
Adding training records is also a good way to improve the style and readability of the code beyond just getting the functional aspects correct. In this example, we want to improve readability and ensure proper precedence by putting each comparison within parentheses. Without this instruction, the LLMs won't generally perform this task automatically.

The result is that:
if (mod.option == 249 && mod.num2 > 96)

gets replaced with:

if ((mod.option == 249) && (mod.num2 > 96))

FmPro Migrator 11.28 also includes the following enhancements:
* Added support for the Google Gemini 2.0-flash model.
* Added support for the Anthropic models.
* When using Anthropic models to convert scripts to FileMaker Pro, the prompt will include prompting to create the FileMaker Script Workspace XML. The clipboard button will put the script onto the clipboard ready for pasting into the Script Workspace. The latest claude-3-5-sonnet-20241022 model works best for creating FileMaker script XML for Visual FoxPro or Microsoft Access to FileMaker Pro conversion projects.
* Improved code output from the models by removing unneeded explanatory text. This change can be overridden by holding down the Shift key when clicking the Convert button.
* Improved the calculation of input token sizes for models.
* Conversions of code into FileMaker Pro now use the phrase "script steps" in order to obtain a better conversion result across all models and vendors.
* Improved saving/restoring of user-entered API keys to enable having separate keys for each AI vendor.
* Fixed an issue importing VFP Forms/Reports which caused some layouts to crash when copying into FileMaker Pro due to an incorrectly formed Tab Control.
* Fixed an issue with some VFP text labels being converted with a black background.
* Fixed an issue with VFP line objects having a thickness > 1 being displayed as diagonal lines.

Availability:
The AI Accelerated Edition of FmPro Migrator is available immediately, and includes up to 500K daily AI processing tokens and the conversion of up to 500 forms/reports/layouts into GUI conversion projects. [Token counts are unlimited when using Ollama hosted models with FmPro Migrator Custom DEV Edition or [FmPro Migrator Site License Edition](.).]

Founded in 1999 by David Simpson in Sunny California, .com Solutions Inc. develops multi-platform graphical applications supporting FileMaker Pro, Visual FoxPro, SQLite, Oracle, MySQL, Access, SQL Server, Sybase, DB2, PostgreSQL, VB6, and LiveCode developers. VFP Code Converter, converts FoxPro scripts into 50+ modern programming languages. FmPro Script Diff, compares FileMaker Pro scripts. SQLite Diff compares SQLite database schema and data. Copyright (C) 1999-2025 .com Solutions Inc. All Rights Reserved. FileMaker is a registered trademark of Claris - an Apple company. Apple, the Apple logo, and macOS platforms are trademarks of Apple Inc. in the U.S. and/or other countries.

David Simpson
.com Solutions Inc.
+1 510-555-1212
email us here
Visit us on social media:
Facebook
X
LinkedIn
Instagram
YouTube
TikTok

This press release can be viewed online at: https://www.einpresswire.com/article/778101052