



Layout Conversion Workbench

Automates layout/form conversions into multiple GUI languages/frameworks.

- Token-efficient metadata supports hundreds of layout objects without exceeding LLM context limits.
- Builds project files ready for immediate development in Claude Code, Cursor, GitHub Copilot & OpenAI tools.
- Local or cloud-based LLM processing.
- Supports modern GUI frameworks via customizable task prompts.

Layout Conversion Workbench Development Window

Src	Version	Conversion Name
dcsi	3.51	.NET XAML WPF + EF Core + C#
dcsi	3.51	.NET XAML WPF + EF Core + VB
dcsi	1.02	ASP .NET C#
dcsi	1.003	Angular + TypeScript
dcsi	1.00	Angular + TypeScript - REST API
dcsi	1.089	CakePHP 5.35
User	1.08	Node.JS

Conversion Name: (Unique) Version: 3.51
.NET XAML WPF + EF Core + C#

System Prompt:
ROLE: Senior .NET Desktop Architect with 15+ years of experience building production WPF applications using XAML, EF Core, and the MVVM pattern. You produce complete, compile-ready C# and XAML files that follow modern .NET 9/10 conventions: CommunityToolkit.Mvvm for ViewModels, dependency injection via Microsoft.Extensions.DependencyInjection, and EF Core for all data access. You excel at visual hierarchy, whitespace management, and mapping legacy form-based UIs into clean WPF layouts. Output files must never contain references to legacy database tools or source platforms - the output should appear to have been written from scratch.

Comments:

Active	Seq.	Type	Task
☒	1	project	getLayoutImages
☒	2	project	.csproj
☒	3	project	.sln
☒	4	project	ValueLists -> AppResources.xaml
☐	5	project	ValueLists -> C# -> AppResources.xaml [Large Number of Tables]
☒	6	project	getAllRelationships & TOs
☒	7	project	Create EF Core Models

Task Description: Sequence: 2 Type: project

Output Filename: App1.csproj

Task Prompt: (Debug)
Generate a modern SDK-style .csproj file for a C# WPF desktop application targeting .NET 9 (net9.0-windows), compatible with Visual Studio 2022/2026. Do not include any XML comments referencing the source platform or migration tools.

Core PropertyGroup:
<OutputType>WinExe</OutputType>
<TargetFramework>net9.0-windows</TargetFramework>
<RootNamespace>App1</RootNamespace>
<AssemblyName>App1</AssemblyName>
<UseWPF>true</UseWPF>
<Nullable>enable</Nullable>
<ImplicitUsings>enable</ImplicitUsings>
<Platforms>AnyCPU;x86</Platforms>

Converted Layout:
<Project Sdk="Microsoft.NET.Sdk">
<!-- BeforeBuild/AfterBuild extensibility uses MSBuild <Target> elements directly in this file. -->
<PropertyGroup>
<OutputType>WinExe</OutputType>
<TargetFramework>net9.0-windows</TargetFramework>
<RootNamespace>App1</RootNamespace>
<AssemblyName>App1</AssemblyName>
<UseWPF>true</UseWPF>
<Nullable>enable</Nullable>
<ImplicitUsings>enable</ImplicitUsings>
<Platforms>AnyCPU;x86</Platforms>
<Prefer32Bit>false</Prefer32Bit>
<WarningLevel>4</WarningLevel>
<FileAlignment>512</FileAlignment>
<GenerateAssemblyInfo>false</GenerateAssemblyInfo>

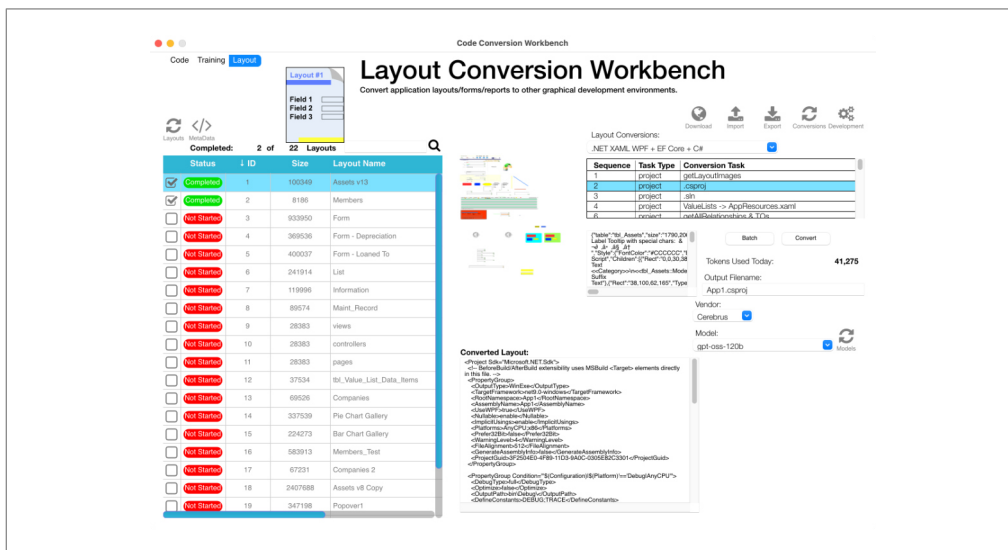
Top Features

- High-fidelity automated GUI conversions across multiple target frameworks and languages.
- Replaces expensive multi-month consulting projects with an automated, cost-effective tool.
- Real-time injection of FileMaker and Visual FoxPro (VFP) metadata directly into task prompts.
- Agentic workflow processes all files and includes batch processing.

Top Features Continued

- Scale up custom apps to use open-source database servers & frameworks, letting local or cloud LLMs handle the repetitive work.
- Extend your team's reach by letting LLMs handle legacy conversions developers don't have time to rewrite manually.
- Especially important for complex conversion projects where the original legacy app developers are no longer available.
- Eliminates project inertia and initial friction when kicking off legacy migration initiatives.
- Rapid adaptability to emerging frameworks, languages, and modern design patterns.
- Saves thousands of dollars and many months of manual application development work.
- Expands your developer pool by migrating legacy systems to widely adopted open-source frameworks.
- Reduce licensing costs for custom apps when scaling up solutions.

Layout Conversion Workbench Screenshot



Layouts shown in the grid display a thumbnail image and stored metadata when clicked. Vendor, LLM model and batch processing options are enabled from the main Layout Conversion Workbench window.

Available & Planned(*) Conversions Include:

.NET XAML WPF + EF Core + C#	.NET XAML WPF + EF Core + VB
*ASP .NET C#	*CakePHP 5.x
*Angular + TypeScript	*Angular + TS + Java SpringBoot
*Node.JS	*Python + Django
*React Native + TypeScript + CLI	*React Native + TypeScript + Expo
*.NET MAUI XAML + MVVM + C#	