

メシウス開発支援ツールの .NET 10 / Visual Studio 2026 最新対応情報

メシウス株式会社

Developer Solutions 事業部

マーケティング部 プロダクトマーケティングチーム 氏家 晋



本日の内容

1

.NET 10 & Visual Studio 2026への対応

2

新バージョンで追加される予定の新機能

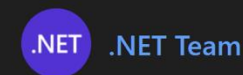
3

新バージョンへの移行方法

.NET 10 リリース

- .NETとしては3回目のLTS
 - 3年間のサポート、修正プログラム提供
 - 2025年11月11日 ~ 2028年11月14日
- .NET 9のサポート終了は2026年11月10日
 - .NET 9からSTSは2年間に変更
 - .NET 8も2026年11月10日でサポート終了

Announcing .NET 10



Today, we are excited to announce the launch of .NET 10, the most productive, modern, secure, intelligent, and performant release of .NET yet. It's the result of another year of effort from thousands of developers around the world. This release includes thousands of performance, security, and functional improvements across the entire .NET stack—from languages and developer tools to workloads—enabling you to build with a unified platform and easily infuse your apps with AI.

Important

.NET 10 is a [Long Term Support \(LTS\) release](#) and will be supported for **three years** until November 10, 2028. We strongly recommend that production applications upgrade to .NET 10 to take advantage of the extended support window, significant performance improvements, and new capabilities.

Downloads of .NET 10 and updates to [Visual Studio 2026](#) and the [C# Dev Kit for Visual Studio Code](#) are available now.

[Download .NET 10](#)

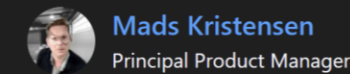
[Get Visual Studio 2026](#)

The .NET team, our partners, and the .NET community are showcasing what's new in .NET 10 at [.NET Conf 2025](#). Watch the [sessions](#) to see all of the excitement including the [keynote](#).

Visual Studio 2026 リリース

- .NET 10をサポートしているVisual Studio
 - Visual Studio 2022は .NET 9まで
- パフォーマンス向上
 - 起動や応答、デバッグ速度がより高速に
- AI支援開発の強化
 - GitHub Copilotの統合

Visual Studio 2026 is here: faster, smarter, and a hit with early adopters



Dear developers,

We're thrilled to announce that Visual Studio 2026 is now generally available! This is a moment we've built side by side with you. Your feedback has helped shape this release more than any before. Since the introduction of the Insiders Channel in September, more developers have downloaded and tested this preview than any other in Visual Studio's history.

In the year leading up to this release, we fixed over 5,000 of your reported bugs and implemented 300 feature requests. That's the most we've ever done, and we're just getting started!

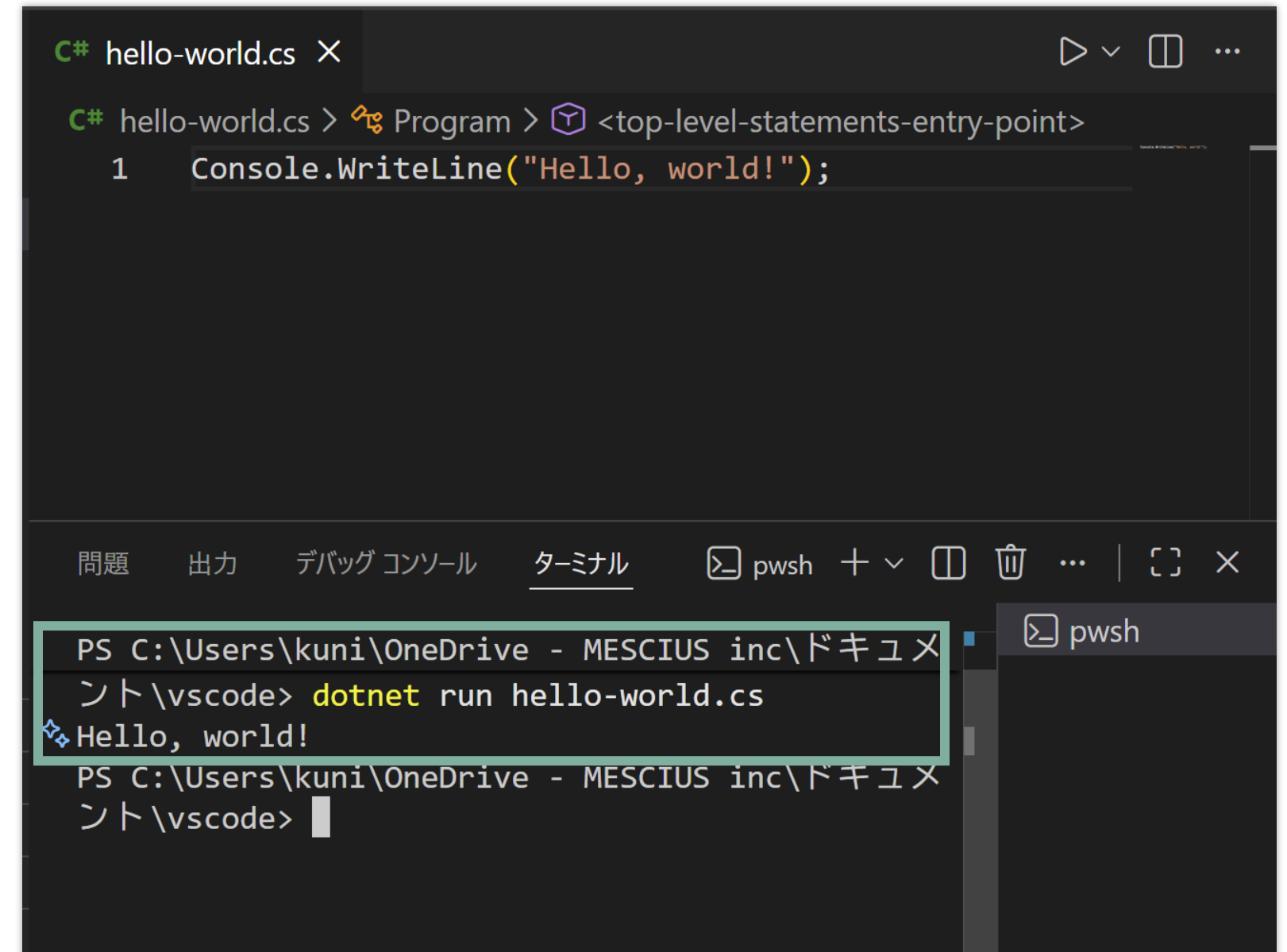
With several performance improvements, a redesigned user experience, and a major leap in AI-driven development, this release is built to help developers focus on what truly matters: creating. Every click, keystroke, and compile is now smoother, faster, and more intuitive – empowering you to turn ideas into code at unprecedented speed.

[Download Visual Studio 2026](#)

You know that sinking feeling when lag interrupts your flow? We've worked hard to make that a thing of the past. **Blazing-fast performance** means startup is significantly snappier, and the UI responds so smoothly you'll barely notice it's there, cutting hangs by over 50% and giving the IDE a lightweight, effortless vibe, even on massive projects. Whether you're wrangling enterprise-scale repos or tinkering on smaller codebases, this sets a new bar for getting stuff done.

気になるポイント（ファイルベースアプリ）

- ファイルベースアプリ（File-based apps）
 - プロジェクトファイル（*.csproj）不要でC#のプログラム（*.cs）を直接実行できる
- Visual Studio Code + C# Dev Kitが必要
- `dotnet run hello-world.cs`で実行可能



```
C# hello-world.cs ×
C# hello-world.cs > Program > <top-level-statements-entry-point>
1 Console.WriteLine("Hello, world!");

問題 出力 デバッグ コンソール ターミナル pwsh + v [] [] ... | [ ] x
PS C:\Users\kuni\OneDrive - MESCIUS inc\ドキュメント\vscode> dotnet run hello-world.cs
Hello, world!
PS C:\Users\kuni\OneDrive - MESCIUS inc\ドキュメント\vscode>
```

気になるポイント（ファイルベースアプリ）

- NuGetパッケージの参照
 - `#:package`を使用
- ASP.NET Core アプリを作成
 - `#:sdk`を使用
- 別のプロジェクトの参照
 - `#:project`を使用

```
C# hello-world-web.cs ×
C# hello-world-web.cs > Program > <top-level-statements-entry-point>
1  #:sdk Microsoft.NET.Sdk.Web
2  #:package Microsoft.AspNetCore.OpenApi@10.*-*
3
4  var builder = WebApplication.CreateBuilder();
5
6  builder.Services.AddOpenApi();
7
8  var app = builder.Build();
9
10 app.MapGet("/", () => "Hello, world!");
11 app.Run();
```

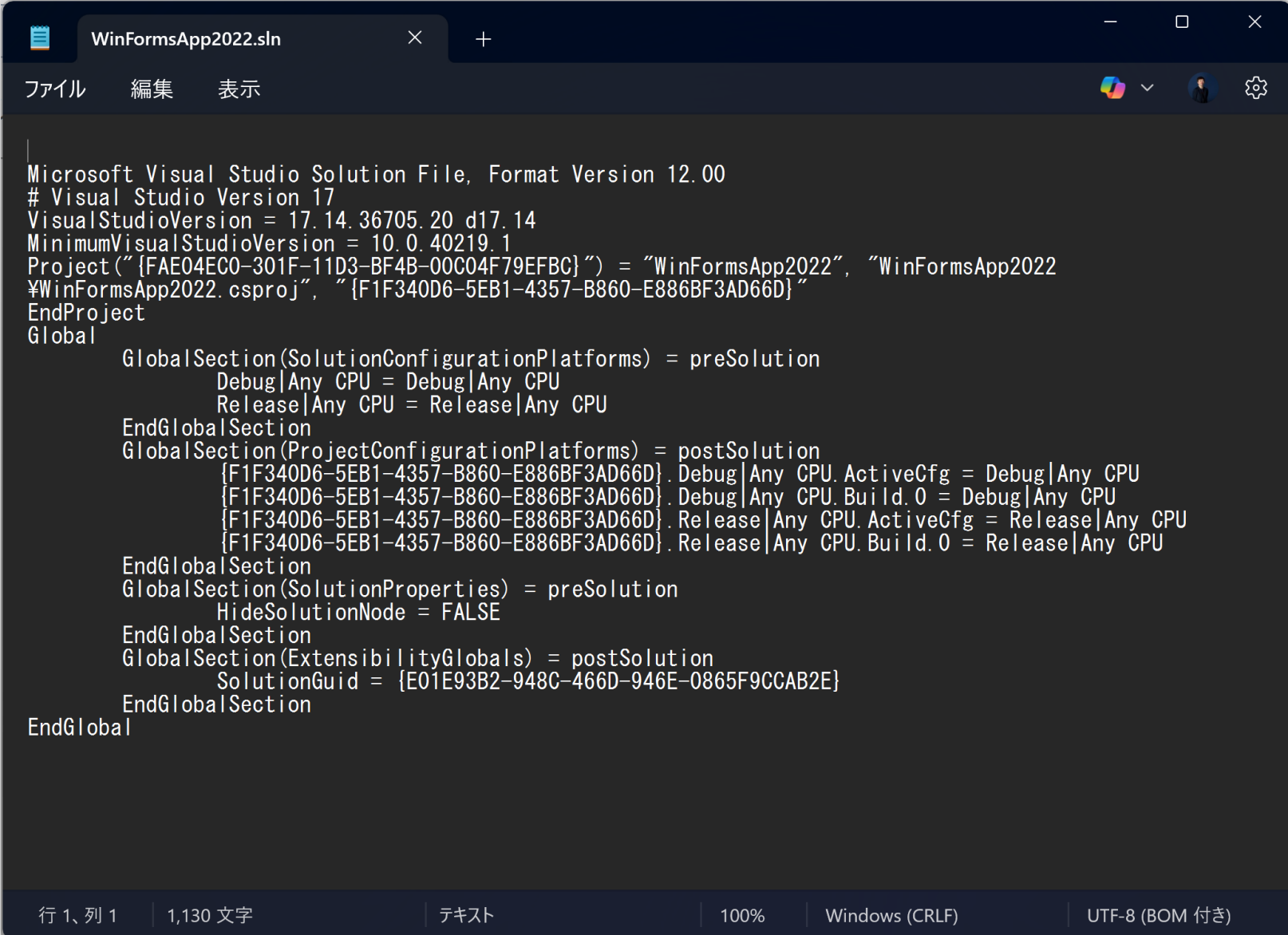
問題 出力 デバッグ コンソール ターミナル

```
PS C:\Users\kuni\OneDrive - MESCIUS inc\ドキュメント\vscode> dotnet run hello-world-web.cs
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...

PS C:\Users\kuni\OneDrive - MESCIUS inc\ドキュメント\vscode> curl localhost:5000
Hello, world!
```

気になるポイント（ソリューションファイル）

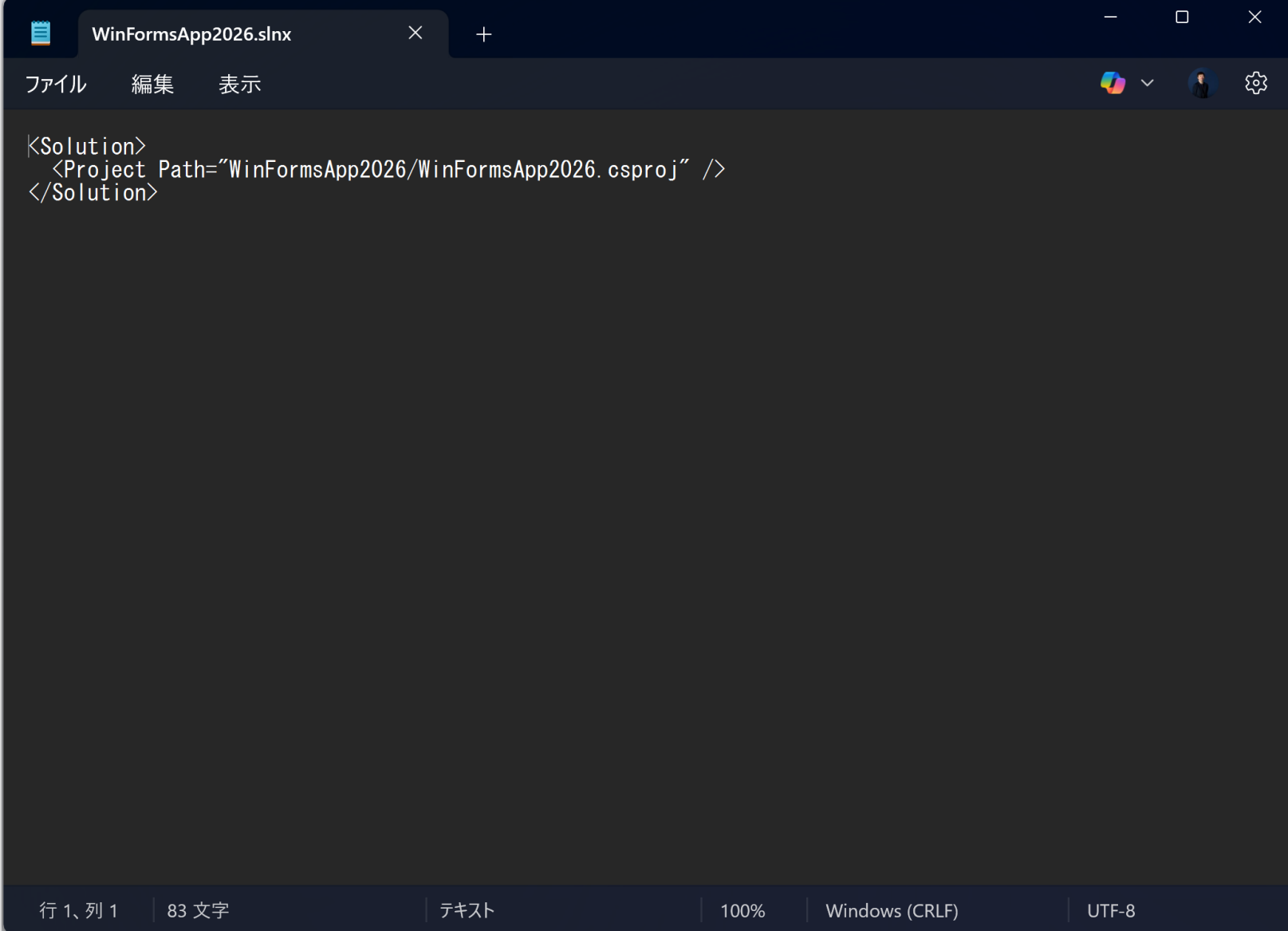
- ソリューションファイル
 - 複数のプロジェクトをまとめるためのファイル
 - ビルドやプラットフォームの情報、GUIDなどが記載
 - Visual Studio 2026ではSLN (*.sln) からSLNX (*.slnx) にデフォルト設定が変更



```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio Version 17
VisualStudioVersion = 17.14.36705.20 d17.14
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "WinFormsApp2022", "WinFormsApp2022
¥WinFormsApp2022.csproj", "{F1F340D6-5EB1-4357-B860-E886BF3AD66D}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {F1F340D6-5EB1-4357-B860-E886BF3AD66D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {F1F340D6-5EB1-4357-B860-E886BF3AD66D}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {F1F340D6-5EB1-4357-B860-E886BF3AD66D}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {F1F340D6-5EB1-4357-B860-E886BF3AD66D}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {E01E93B2-948C-466D-946E-0865F9CCAB2E}
    EndGlobalSection
EndGlobal
```

気になるポイント（ソリューションファイル）

- SLNXはXMLベース
 - 変更箇所の把握やバージョン管理が容易に
- `dotnet sln migrate`でSLNXへ移行可能
 - 元のSLNファイルはそのまま残る



```
<Solution>
  <Project Path="WinFormsApp2026/WinFormsApp2026.csproj" />
</Solution>
```

The screenshot shows a code editor window titled "WinFormsApp2026.slnx". The editor displays the XML content of the solution file. The status bar at the bottom indicates "行 1、列 1" (Line 1, Column 1), "83 文字" (83 characters), "テキスト" (Text), "100%", "Windows (CRLF)", and "UTF-8".

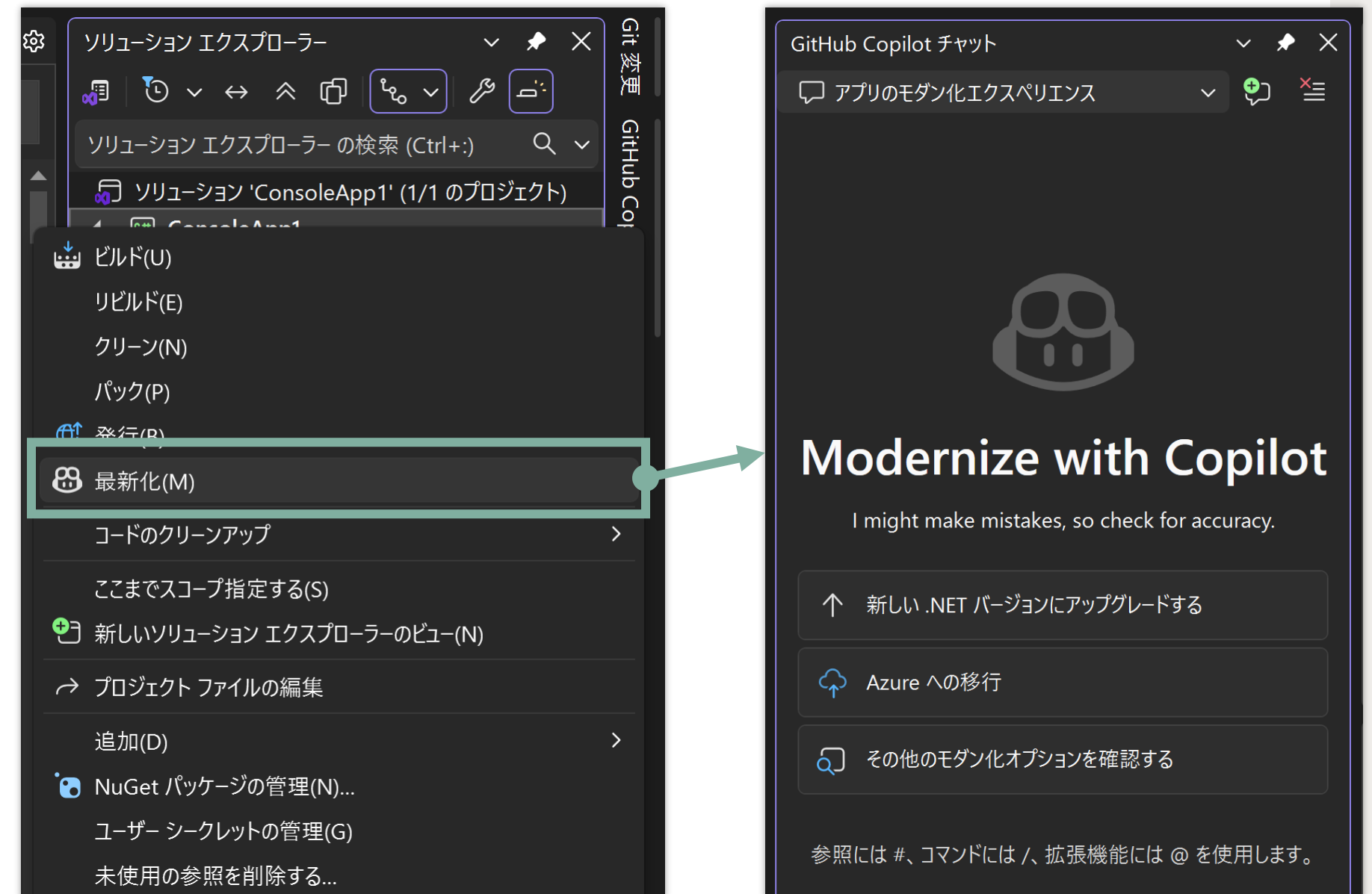
気になるポイント（ソリューションファイル）

- 今までどおりSLNをデフォルトで使用するように設定することも可能

The screenshot shows the Visual Studio settings interface. On the left, a navigation pane lists various settings categories under 'すべての設定' (All Settings). The 'プロジェクトとソリューション' (Projects and Solutions) category is expanded, and the '全般' (General) sub-category is selected. On the right, the 'すべての設定' (All Settings) pane shows the 'プロジェクトとソリューション' (Projects and Solutions) section. Under the '全般' (General) sub-section, several checkboxes are checked, including 'ビルドがエラーで終了した場合は常にエラー一覧を表示する' (Always show error list when build ends with error), 'ビルド開始時に出カウィンドウを表示' (Show output window when build starts), and 'ファイル名の変更時のシンボルの名前変更を有効にする' (Enable symbol name change when file name changes). At the bottom of the settings pane, the '既定のソリューション ファイル形式' (Default Solution File Format) is highlighted with a red box. It is currently set to '.slnx (XML ソリューション ファイル形式)' (Default Solution File Format). Below this, a text box explains: '新しく作成されたソリューションのために使用するファイル形式を定義します。' (Define the file format to use for newly created solutions).

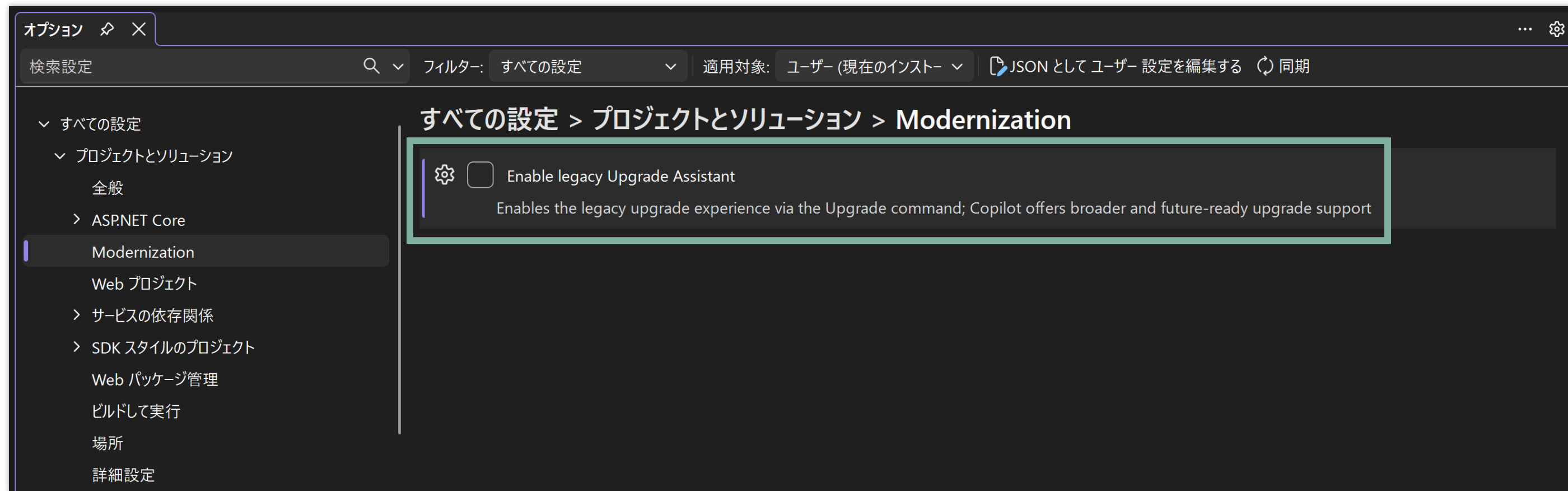
気になるポイント（.NET アップグレードアシスタント）

- Visual Studio 2026ではデフォルトで使用できない状態
- GitHub Copilot アプリを使用して移行するように設定されている
- GitHub CopilotはPro以上の契約が必要

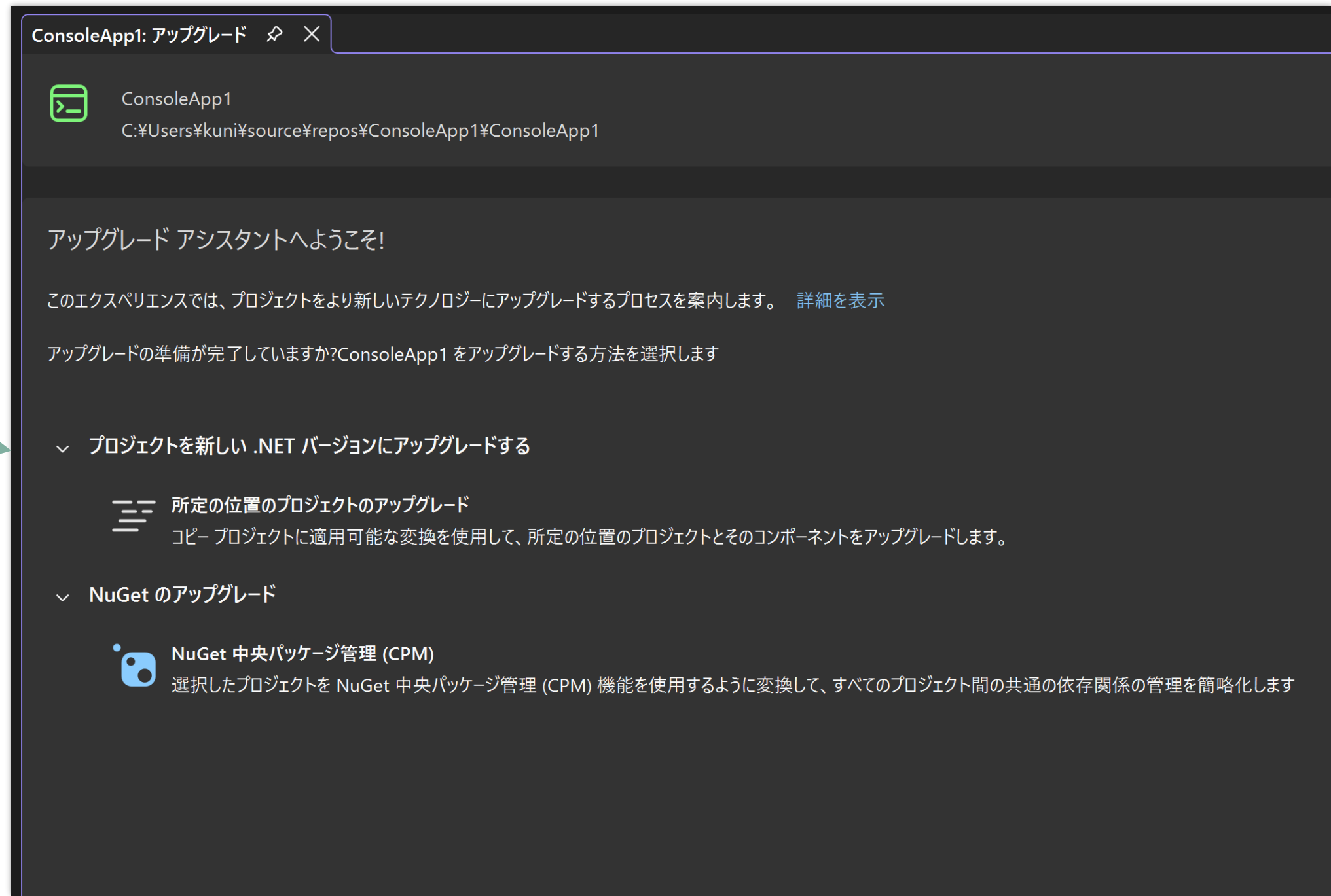
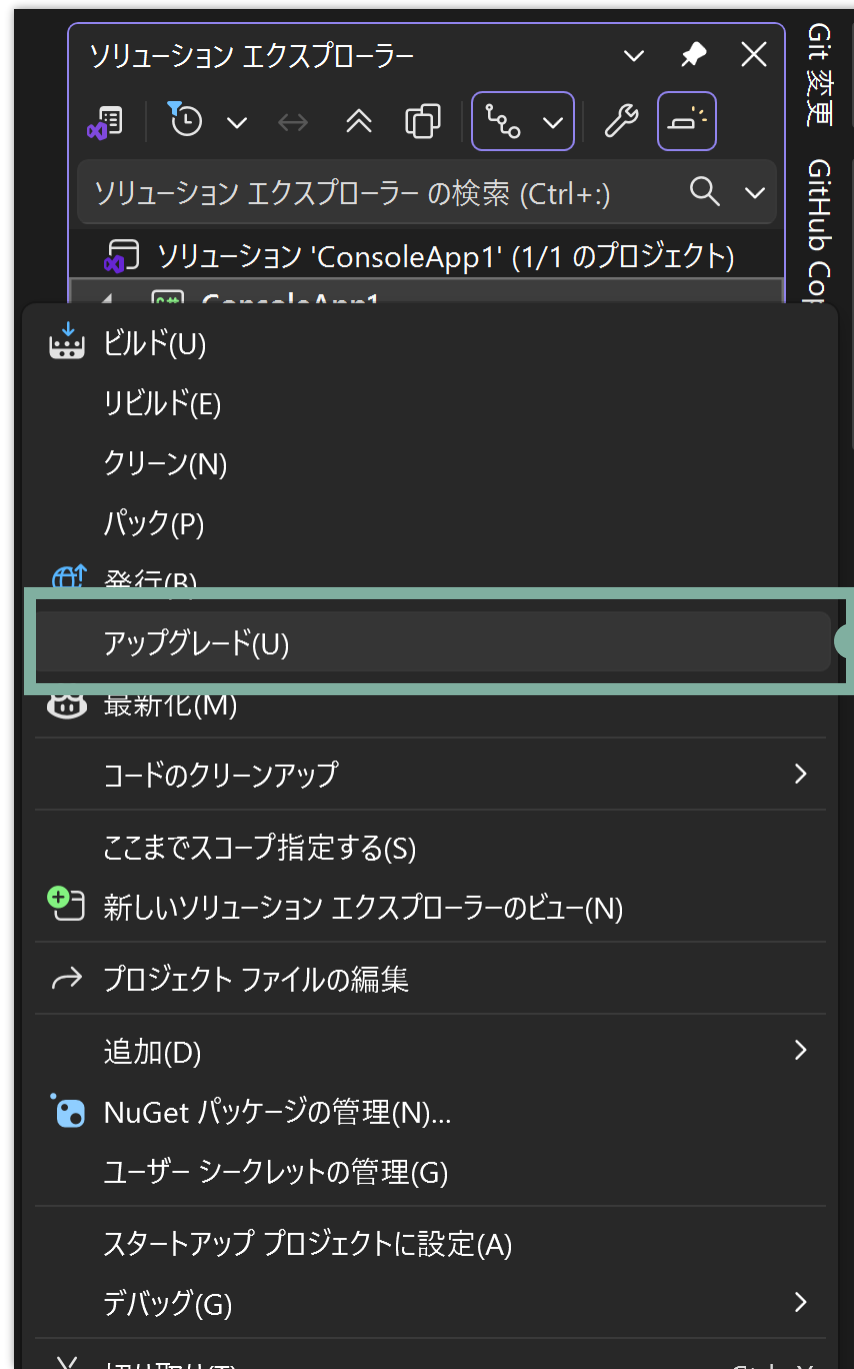


気になるポイント（.NET アップグレードアシスタント）

- 使用するには以下の設定の変更が必要



気になるポイント (.NET アップグレードアシスタント)

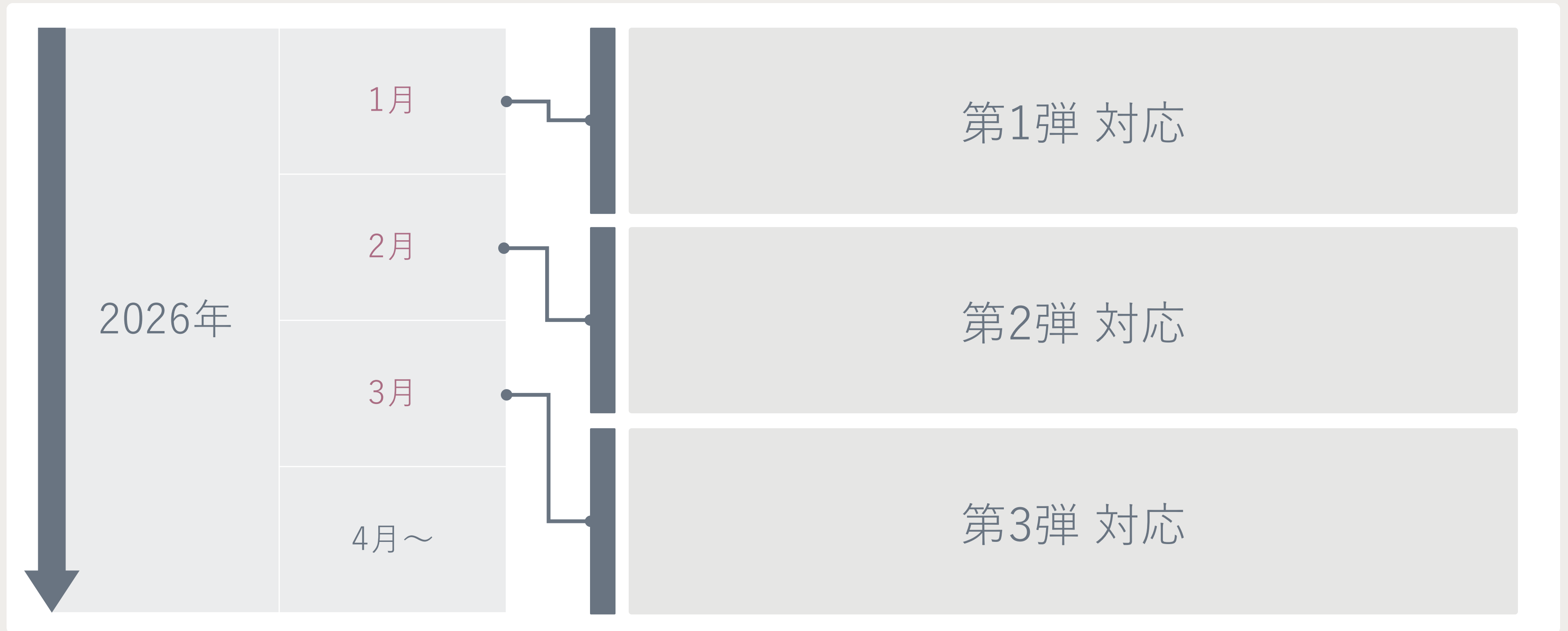


.NET 10 & Visual Studio 2026への対応

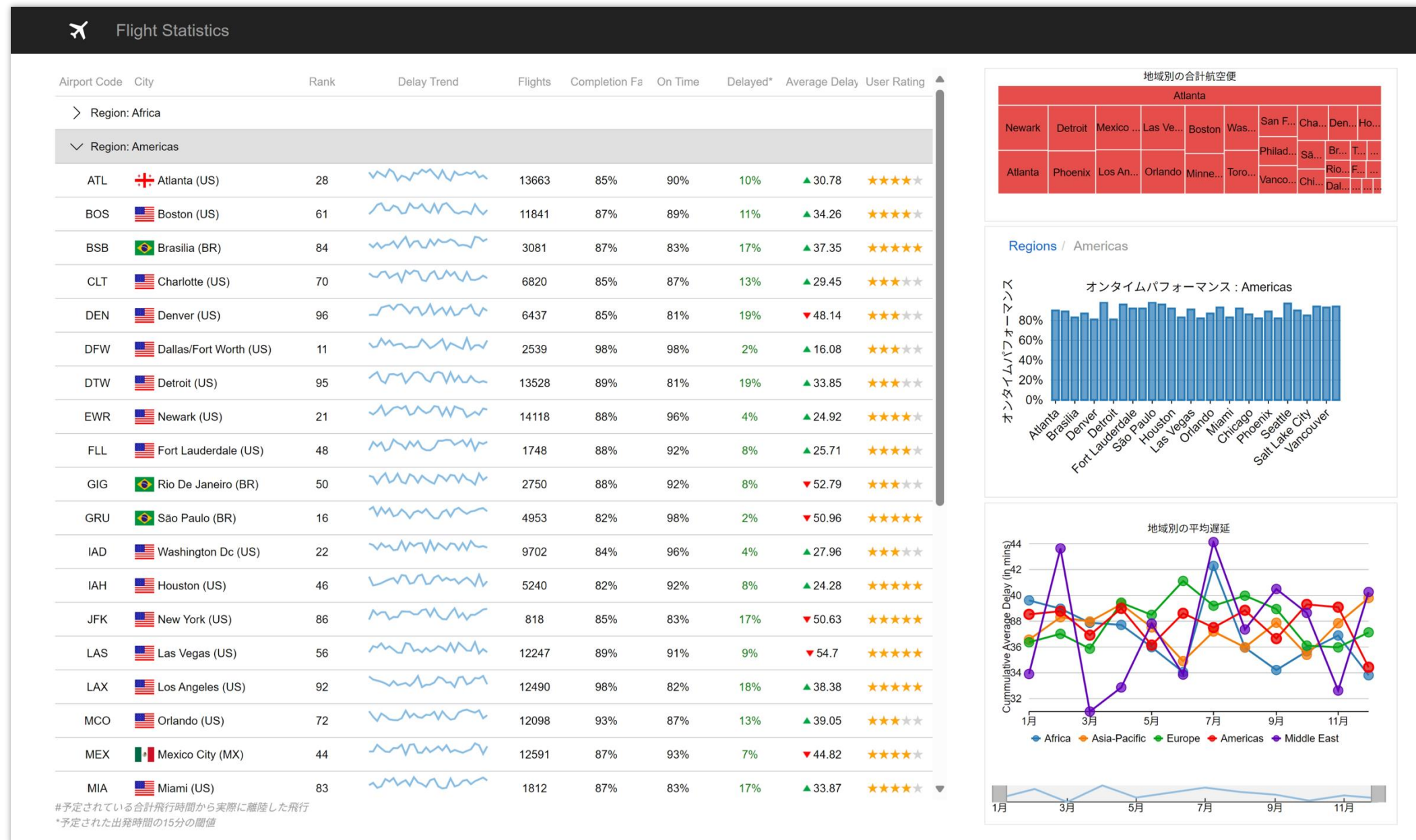
- .NET Framework や .NET 8 からの移行もしくは .NET 10 での新規開発をご検討のみなさまにメシウス製品の .NET 10 と Visual Studio 2026 対応のロードマップをお伝えします。



.NET 10 & Visual Studio 2026対応のロードマップ



.NET 10 & Visual Studio 2026対応 (2026年1月)

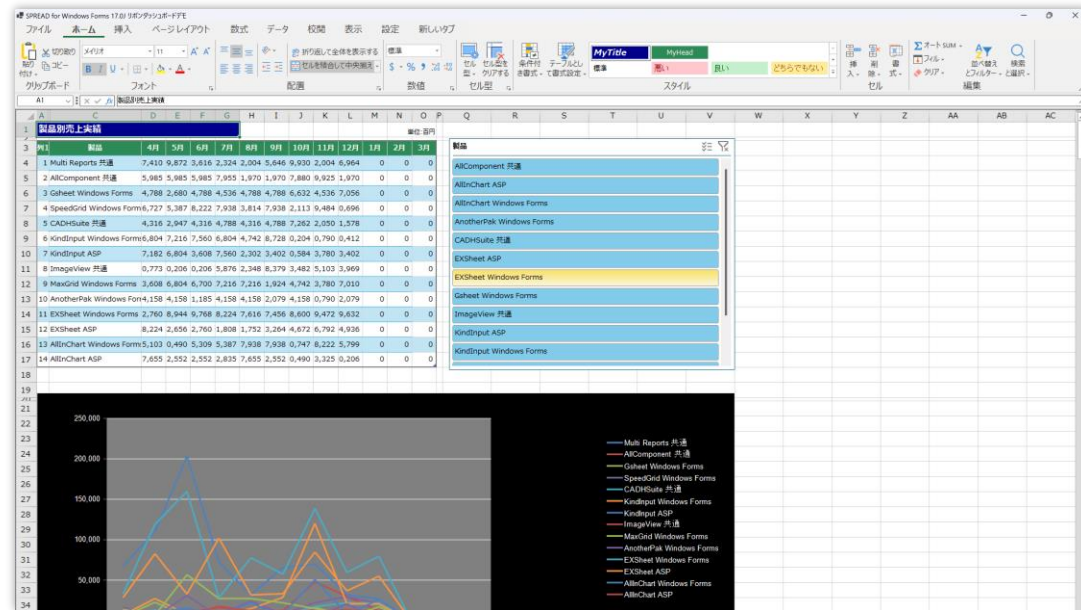


.NET開発用UIコンポーネント
セット

ComponentOne

.NET 10 & Visual Studio 2026対応 (2026年2月~3月)

Excelライクなスプレッドシート



SPREAD

for Windows Forms

入力支援コントロールセット



InputManPlus

for Windows Forms

多段明細データグリッド

商品コード	区分名	商品名		単価	生産中止	仕入先コード		仕入先名	
		区分コード	説明			発注点	都道府県	電話番号	住所1
1	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						
2	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						
3	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						
4	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						
5	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						
6	1 飲料	果汁100% オレンジ	清涼飲料、コーヒー、紅茶、ビール、エ	200	100	東京都	目黒区東ヶ丘 1-1-XX	(03)3410-48XX	(03)3410-48XX
			200g×12瓶						

MultiRowPlus

for Windows Forms

.NET 10 & Visual Studio 2026対応 (2026年3月～)

Excel / PDFファイル进行操作するAPIライブラリ

```
// ワークブックの作成
Workbook workbook = new Workbook();

// ワークシートの取得
Worksheet worksheet = workbook.ActiveSheet;

// セル範囲を指定して文字列を設定
worksheet.Range["C3"].Value =
    "こんにちは、世界!DioDocs (ディオドック) です。";

// Excelファイルとして保存
workbook.Save("HelloWorld.xlsx");
```

```
// ドキュメントを生成
GcPdfDocument doc = new GcPdfDocument();

// ページを追加してグラフィックオブジェクトを取得
GcPdfGraphics g = doc.NewPage().Graphics;

// 文字列を描画
g.DrawString("こんにちは、世界!DioDocs (ディオドック) です。",
    new TextFormat() {
        FontName = "Yu Gothic",
        FontSize = 18,
        ForeColor = Color.FromArgb(0, 123, 138)
    },
    new PointF(72, 72));

// PDFファイルとして保存
doc.Save("HelloWorld.pdf");
```

DioDocs

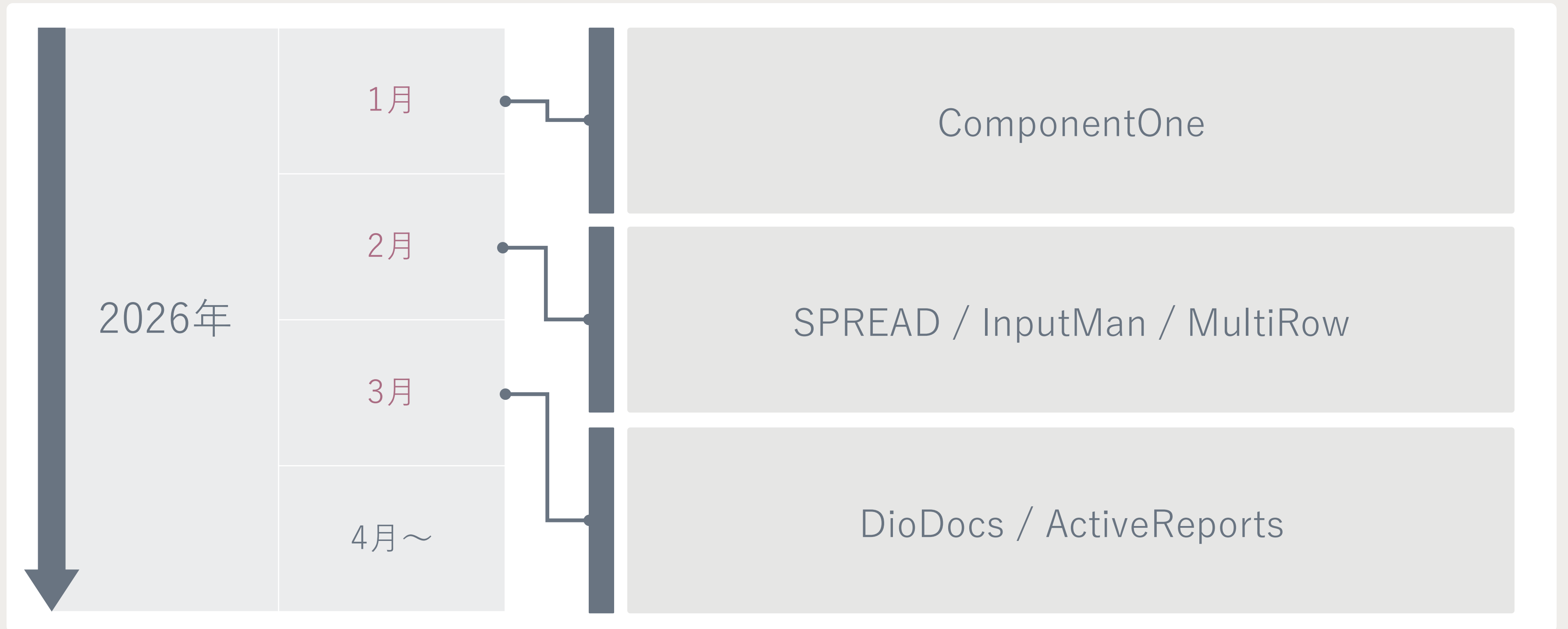
for Excel / PDF

帳票開発コンポーネント

ActiveReports

for .NET

.NET 10 & Visual Studio 2026対応のロードマップ



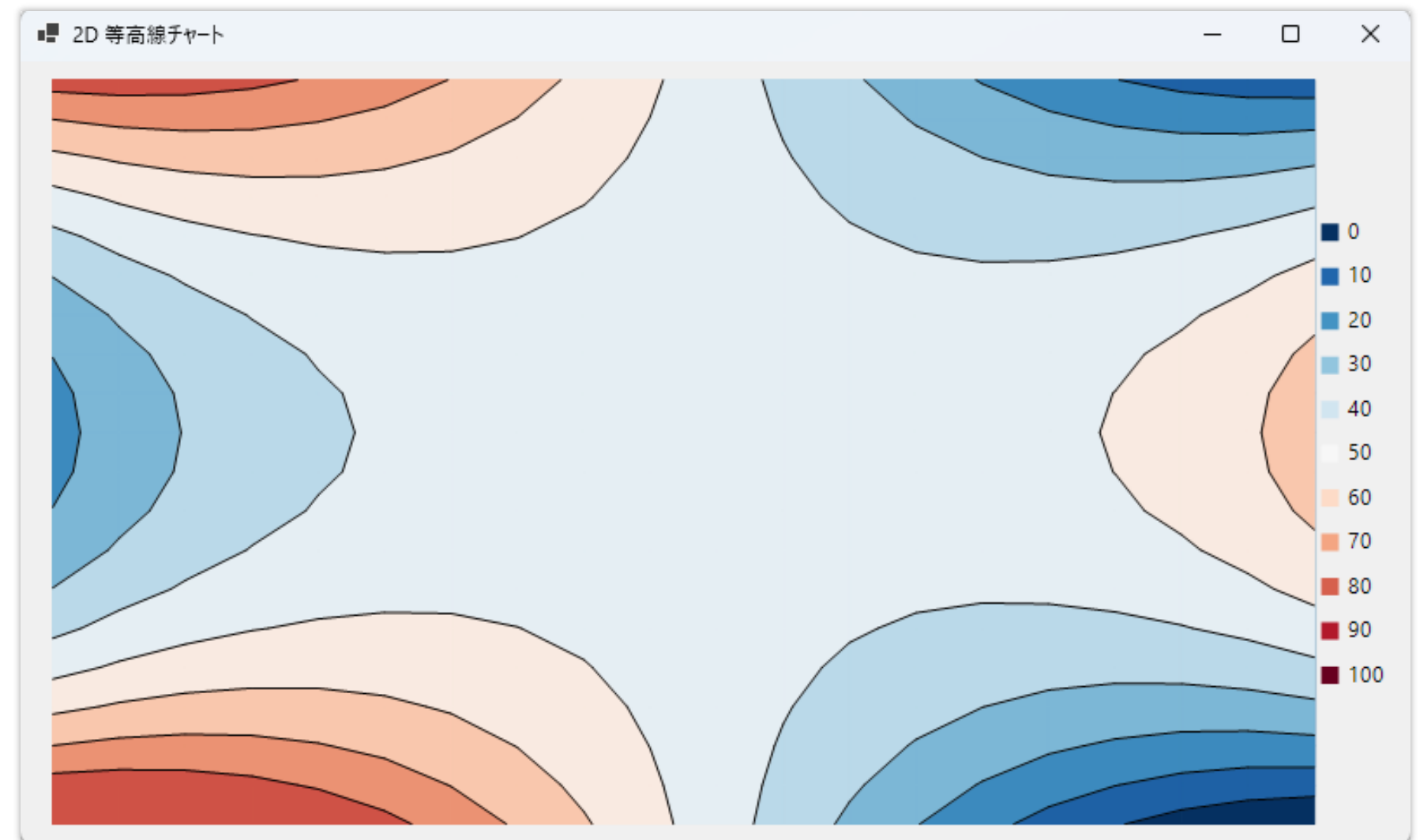
新バージョンで追加される予定の新機能

- ComponentOne 2026J v1
- SPREAD for Windows Forms 19.0J
- InputManPlus / MultiRowPlus for Windows Forms 13.0J



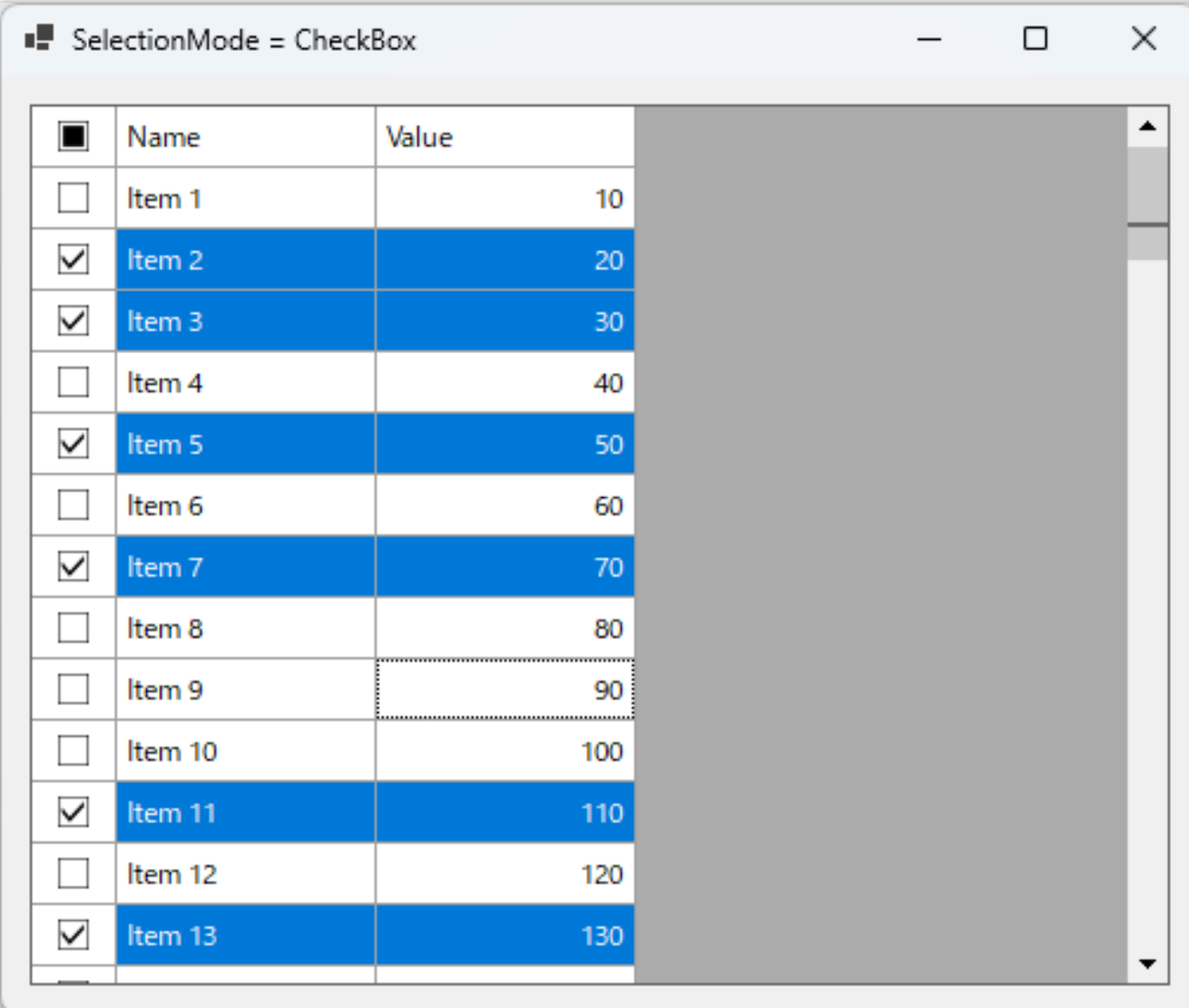
ComponentOne 2026J v1

- FlexChart
 - 2次元等高線チャート
 - 2次元データにおいてデータの大小を色分けして描画



ComponentOne 2026J v1

- FlexGrid
 - チェックボックスによる行の選択
 - すべての行を一括で選択あるいは選択解除することも可能



<input checked="" type="checkbox"/>	Name	Value
<input type="checkbox"/>	Item 1	10
<input checked="" type="checkbox"/>	Item 2	20
<input checked="" type="checkbox"/>	Item 3	30
<input type="checkbox"/>	Item 4	40
<input checked="" type="checkbox"/>	Item 5	50
<input type="checkbox"/>	Item 6	60
<input checked="" type="checkbox"/>	Item 7	70
<input type="checkbox"/>	Item 8	80
<input type="checkbox"/>	Item 9	90
<input type="checkbox"/>	Item 10	100
<input checked="" type="checkbox"/>	Item 11	110
<input type="checkbox"/>	Item 12	120
<input checked="" type="checkbox"/>	Item 13	130

SPREAD for Windows Forms 19.0J

- チャートの機能強化
 - 折れ線チャートのスムージング
 - データテーブル
- 行と列のピン止め
 - ビュー内の行や列をピン止めして常に最前面に表示
- シートタブのスタイル設定
 - シートのロック時の色、ホバー色、アイコンの設定
- 数式エディタパネル
 - 複雑な数式を複数行にわたりインデント付きで表示
- 新しい関数
 - TRIMRANGE、REGEXTTEST、REGEXEXTRACT、REGEXREPLACE

InputManPlus / MultiRowPlus for Windows Forms 13.0J

- JPAAddress
 - 郵便番号・デジタルアドレスAPIによる住所データの取得
- InputMan
 - GcMask / GcCharMaskの書式設定にデジタルアドレスの設定を追加

新サービス「デジタルアドレス」で、
住所全文が取得可能に。

1つのAPIで、デジタルアドレスにも対応

デジタルアドレスとは、住所を「7桁の英数字」で表現できる、
日本郵便が提供する新しいサービスです。

「郵便番号」の代わりに「デジタルアドレス」を入力することで、
住所全文（郵便番号を含む都道府県から町域、建物情報などを含む）の
取得が可能になります。

> [デジタルアドレスについて](#)



デジタルアドレス

@ A B C 1 2 D 6

▼ API

〒100-0004
東京都千代田区大手町
2丁目3-1
JPマンション301



郵便番号と
同じAPIで導入可能



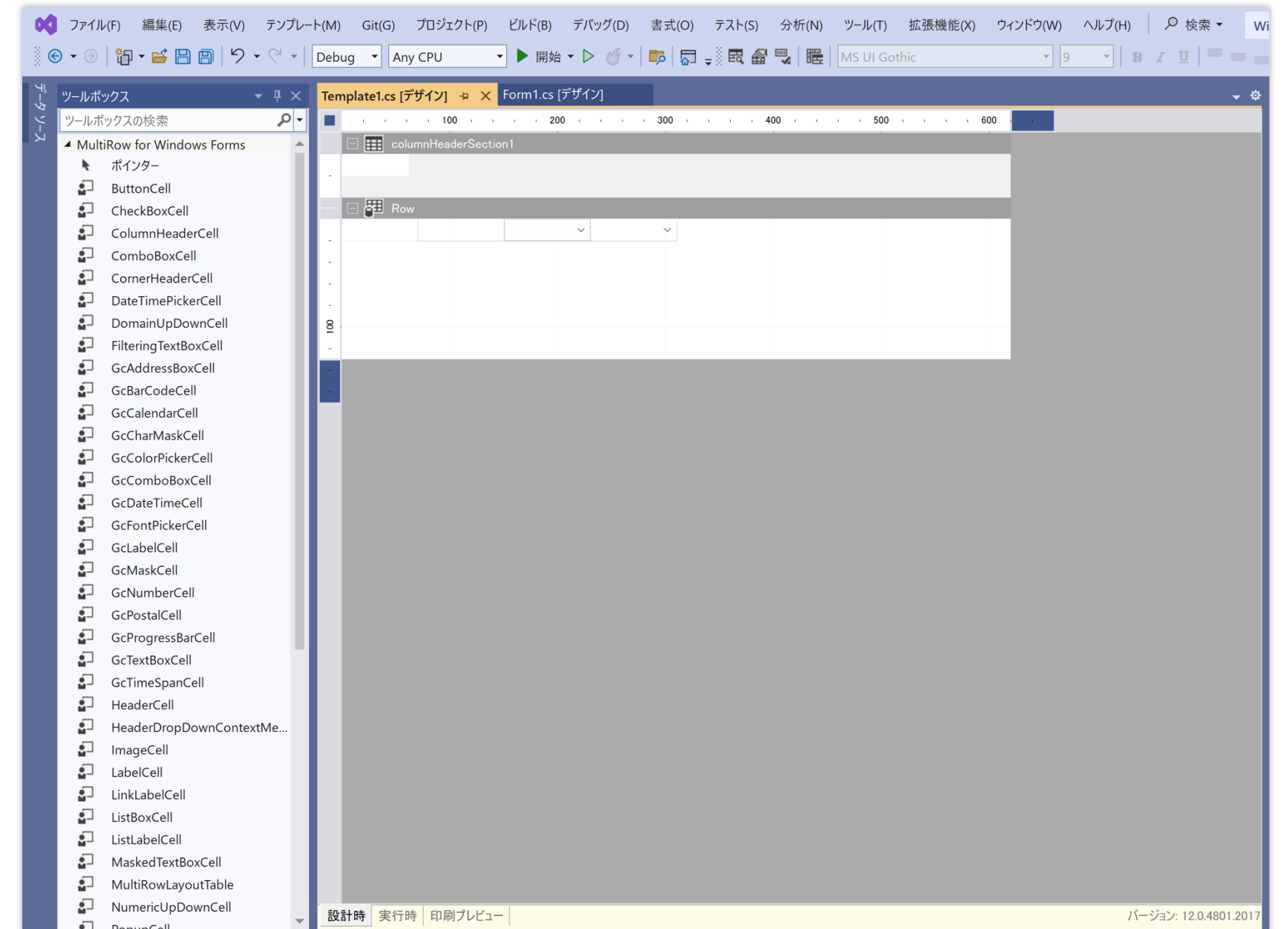
建物情報などの
住所全文取得



複数サービスの
名寄せに貢献

InputManPlus / MultiRowPlus for Windows Forms 13.0J

- MultiRow
 - テンプレートデザイナーの .NET対応
 - .NET Frameworkのプロジェクトが不要で利用できる



デモ (SPREAD for Windows Forms)

バージョンアップの必要性・メリット

- 最新の開発環境で利用できる
 - .NET 10、Visual Studio 2026
- パフォーマンスの向上
 - .NETの改善メリットを享受できる
- .NET 8/9のサポート終了 (2026/11/10)
 - .NET 10に対応したバージョンへ
- Windows 10のサポート終了
 - ~~22H2 (2025/10/14)~~
 - [ESU](#) (2028/10/14)
 - [LTSC](#) (2027/1/12)
- 新機能を利用できる
- 保守サービス契約期間内なら別途費用が不要でバージョンアップ可

バージョンアップの必要性・メリット

- Spread.NET v18.2 (英語版)
 - Excelファイルの読み込みとセルのスタイル設定の処理時間を測定
 - .NET 9 からさらにパフォーマンス向上

BenchmarkDotNet v0.15.6, Windows 11 (10.0.26100.6899/24H2/2024Update/HudsonValley)
 11th Gen Intel Core i7-1195G7 2.90GHz (Max: 2.92GHz), 1 CPU, 8 logical and 4 physical cores
 .NET SDK 10.0.100
 [Host] : .NET 9.0.11 (9.0.11, 9.0.1125.51716), X64 RyuJIT x86-64-v4
 Job-CVJJHE : .NET 9.0.11 (9.0.11, 9.0.1125.51716), X64 RyuJIT x86-64-v4

InvocationCount=1 IterationCount=10 UnrollFactor=1

Method	Mean	Error	StdDev	Gen0	Gen1
OpenExcel	3,842.8 ms	536.39 ms	354.79 ms	59000.0000	30000.0000
SetStyle	944.1 ms	89.18 ms	58.99 ms	43000.0000	-

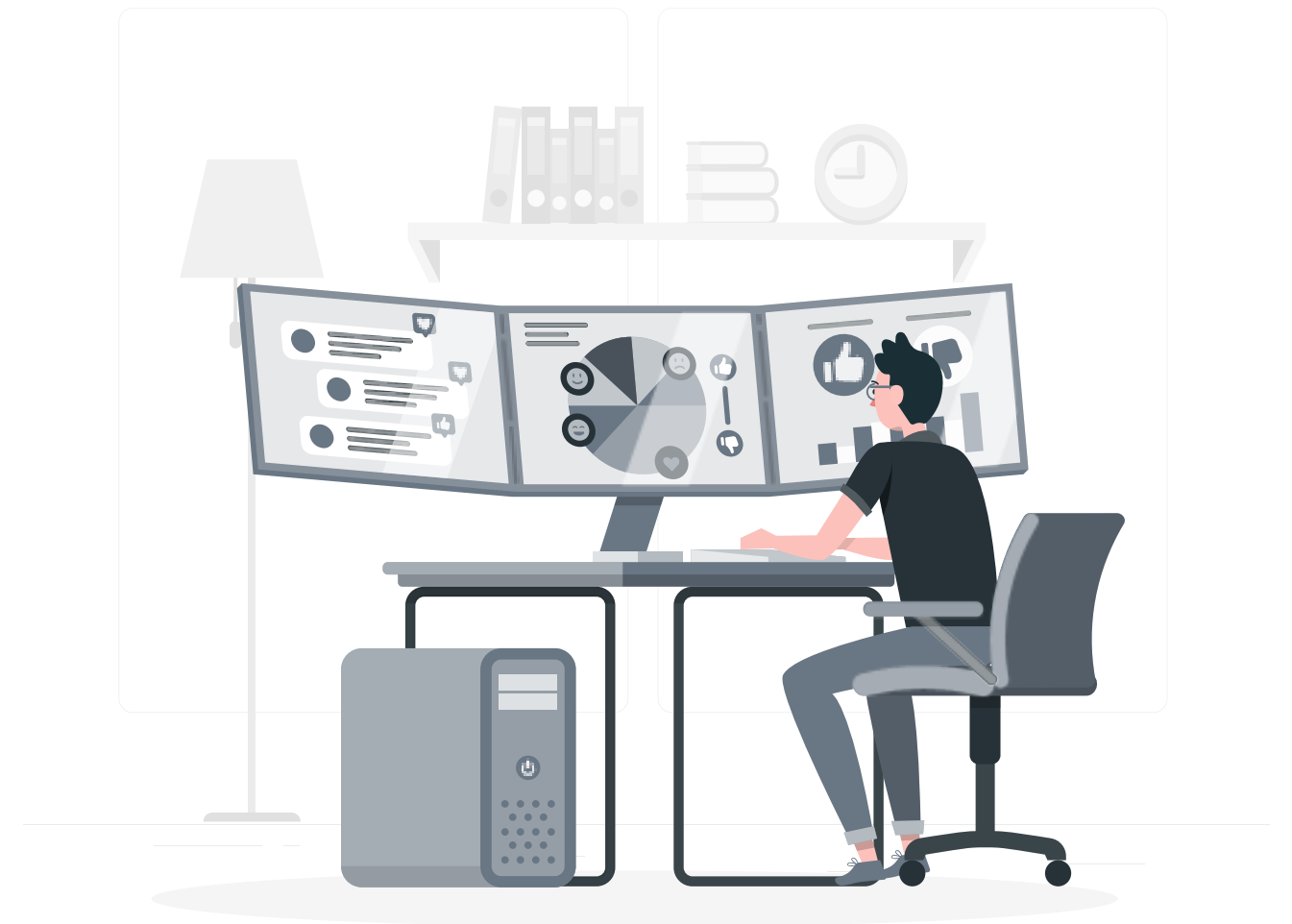
BenchmarkDotNet v0.15.6, Windows 11 (10.0.26100.6899/24H2/2024Update/HudsonValley)
 11th Gen Intel Core i7-1195G7 2.90GHz (Max: 2.92GHz), 1 CPU, 8 logical and 4 physical cores
 .NET SDK 10.0.100
 [Host] : .NET 10.0.0 (10.0.0, 10.0.25.52411), X64 RyuJIT x86-64-v4
 Job-CVJJHE : .NET 10.0.0 (10.0.0, 10.0.25.52411), X64 RyuJIT x86-64-v4

InvocationCount=1 IterationCount=10 UnrollFactor=1

Method	Mean	Error	StdDev	Gen0	Gen1
OpenExcel	3,606.0 ms	288.46 ms	171.66 ms	59000.0000	30000.0000
SetStyle	884.1 ms	40.32 ms	23.99 ms	43000.0000	-

新バージョンへの移行方法

- ComponentOne
- SPREAD / InputManPlus / MultiRowPlus

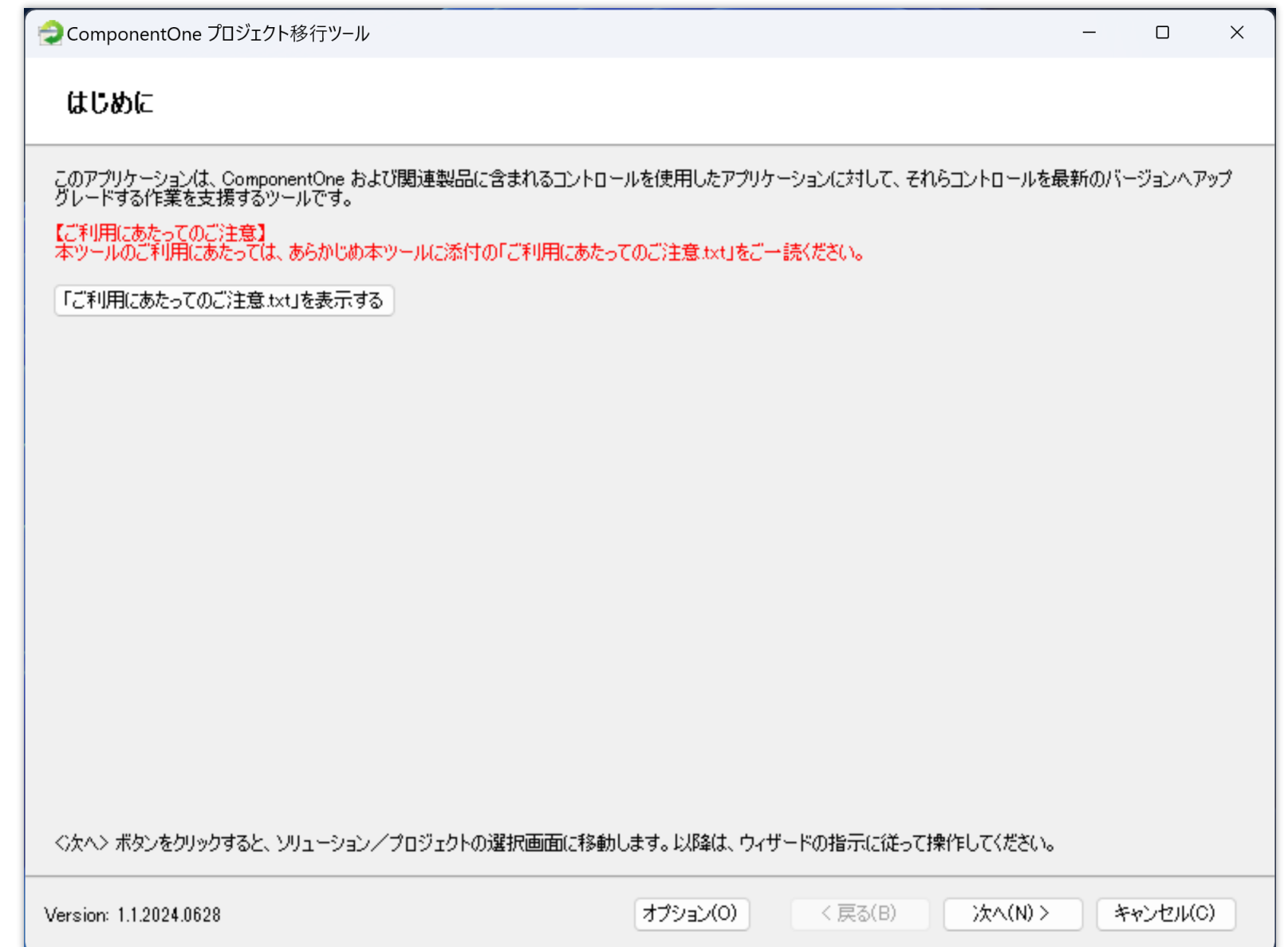


ComponentOne

- アセンブリ参照を使用しているプロジェクト
 - .NET Framework
- NuGet パッケージを使用しているプロジェクト
 - .NET/.NET Framework

アセンブリ参照を使用しているプロジェクト

1. プロジェクト移行ツールを利用
 - Windows Forms/WPF/ASP.NET Web Forms
 - 上記以外では手動で参照設定を更新
2. .NET Frameworkのバージョンを変更
 - プロジェクトファイルを更新
3. 更新履歴で変更点を確認 & 動作を確認



NuGet パッケージを使用しているプロジェクト

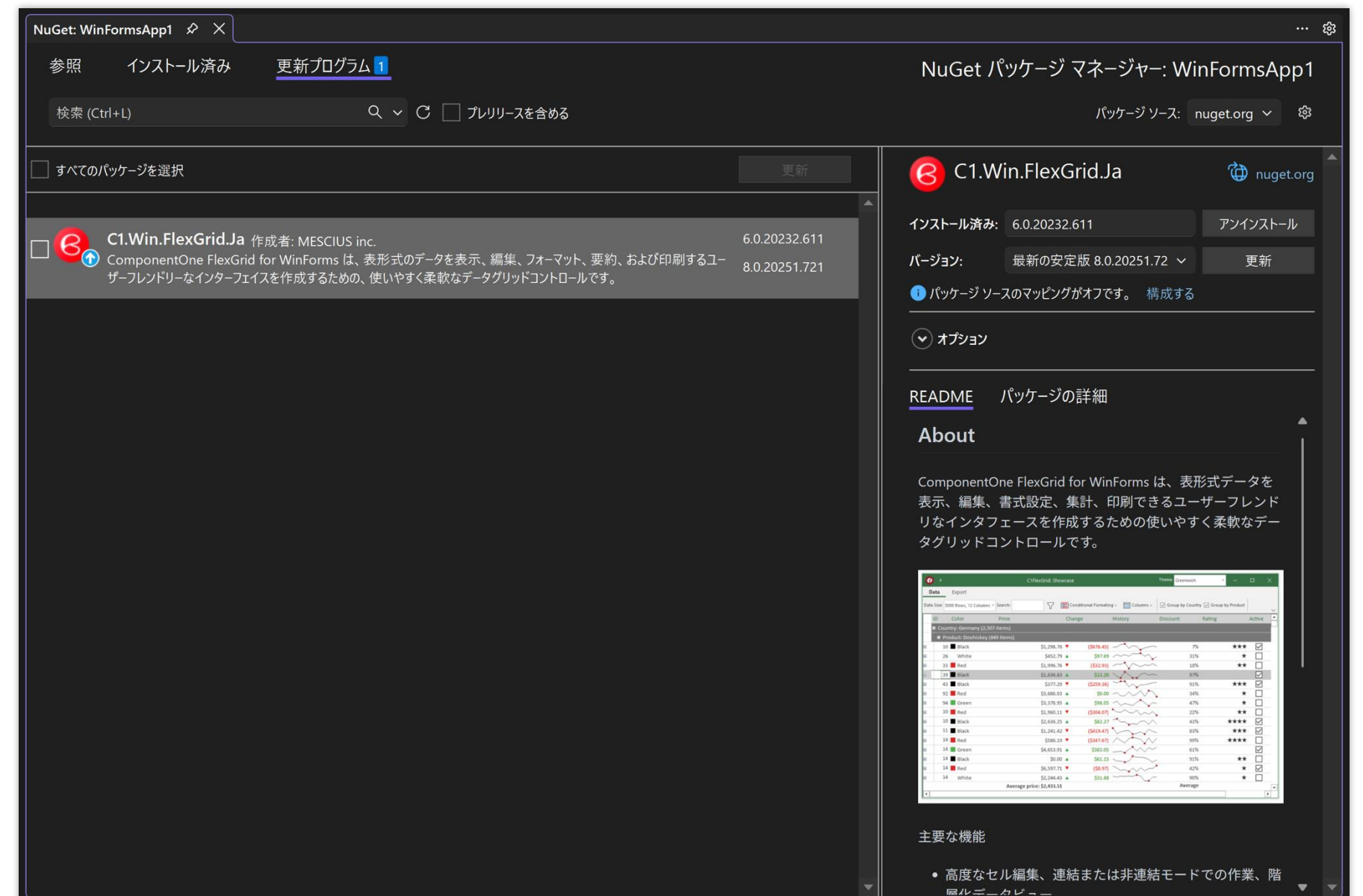
1. NuGetパッケージを更新

- 旧バージョン → 新バージョン

2. .NET / .NET Frameworkのバージョンを変更

- プロジェクトファイルを更新

3. 更新履歴で変更点を確認 & 動作を確認

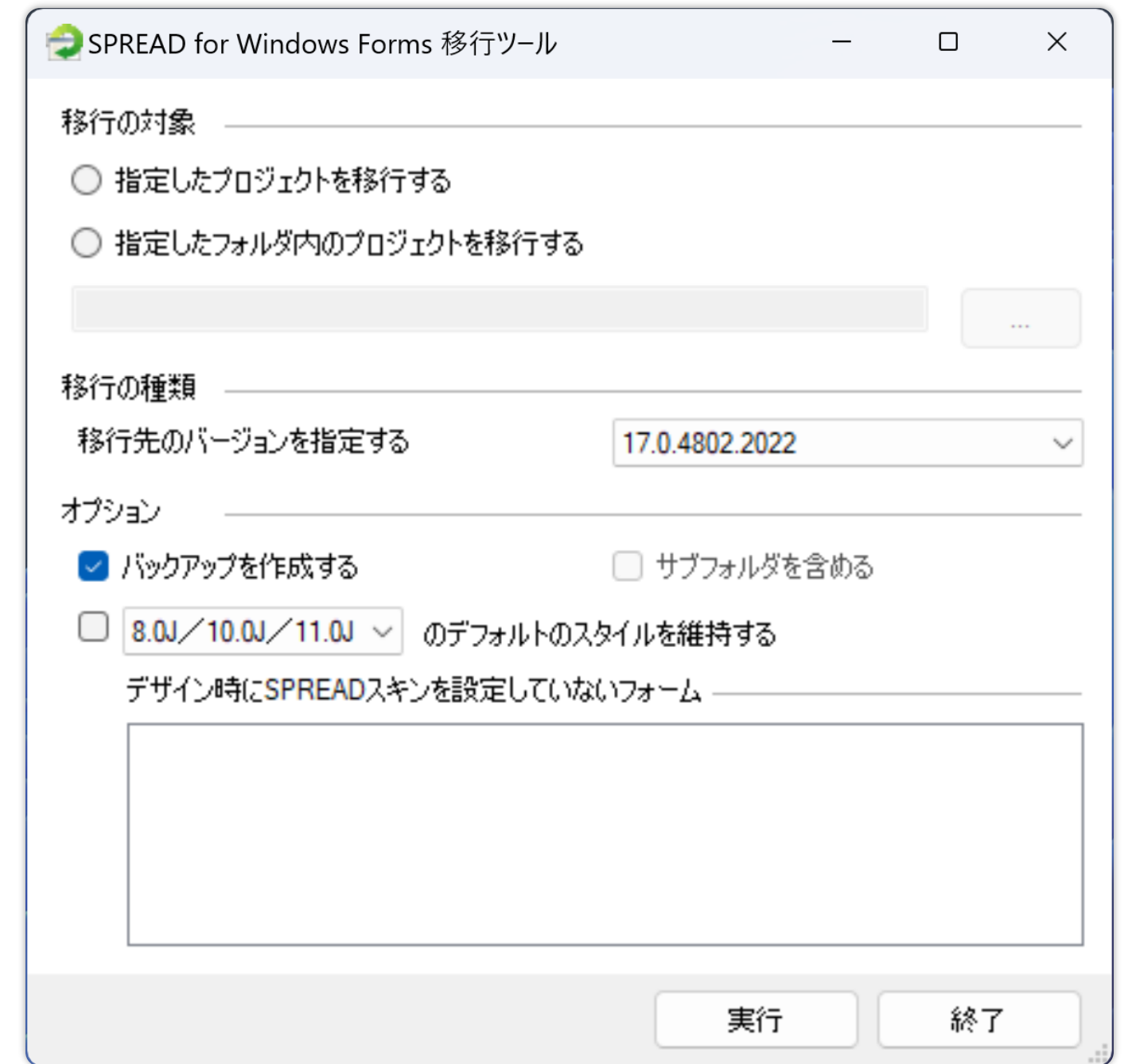


SPREAD / InputManPlus / MultiRowPlus

- アセンブリ参照を使用しているプロジェクト
 - .NET Framework
- NuGet パッケージを使用しているプロジェクト
 - .NET/.NET Framework

アセンブリ参照を使用しているプロジェクト

1. 移行ツールを使用
2. .NET Frameworkのバージョンを変更
 - プロジェクトファイルを更新
3. 更新履歴で変更点を確認 & 動作を確認



NuGet パッケージを使用しているプロジェクト

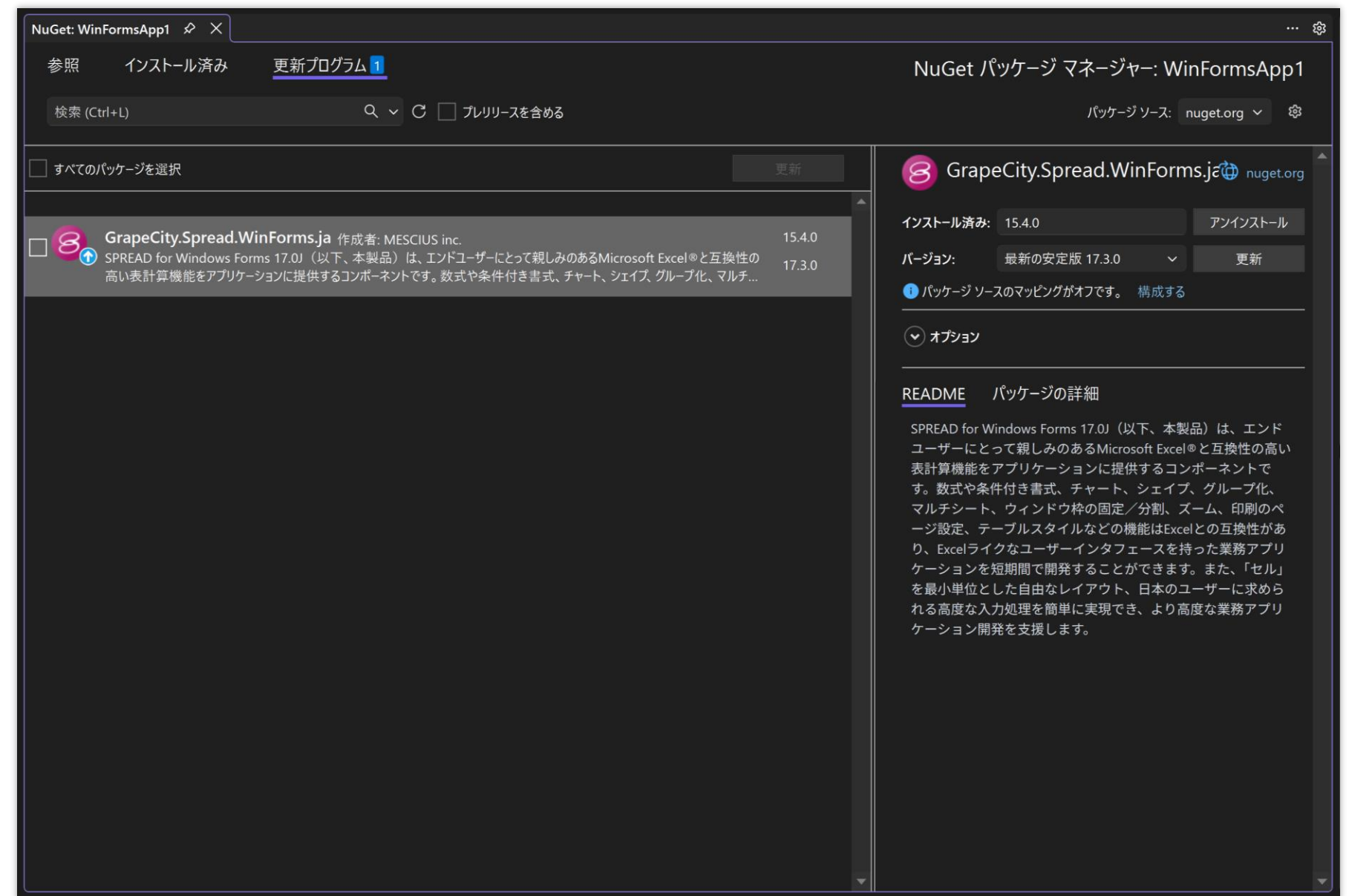
1. NuGetパッケージを更新

- 旧バージョン → 新バージョン

2. .NET / .NET Framework のバージョンを変更

- プロジェクトファイルを更新

3. 更新履歴で変更点を確認 & 動作を確認



.NET Framework から .NET 10への移行

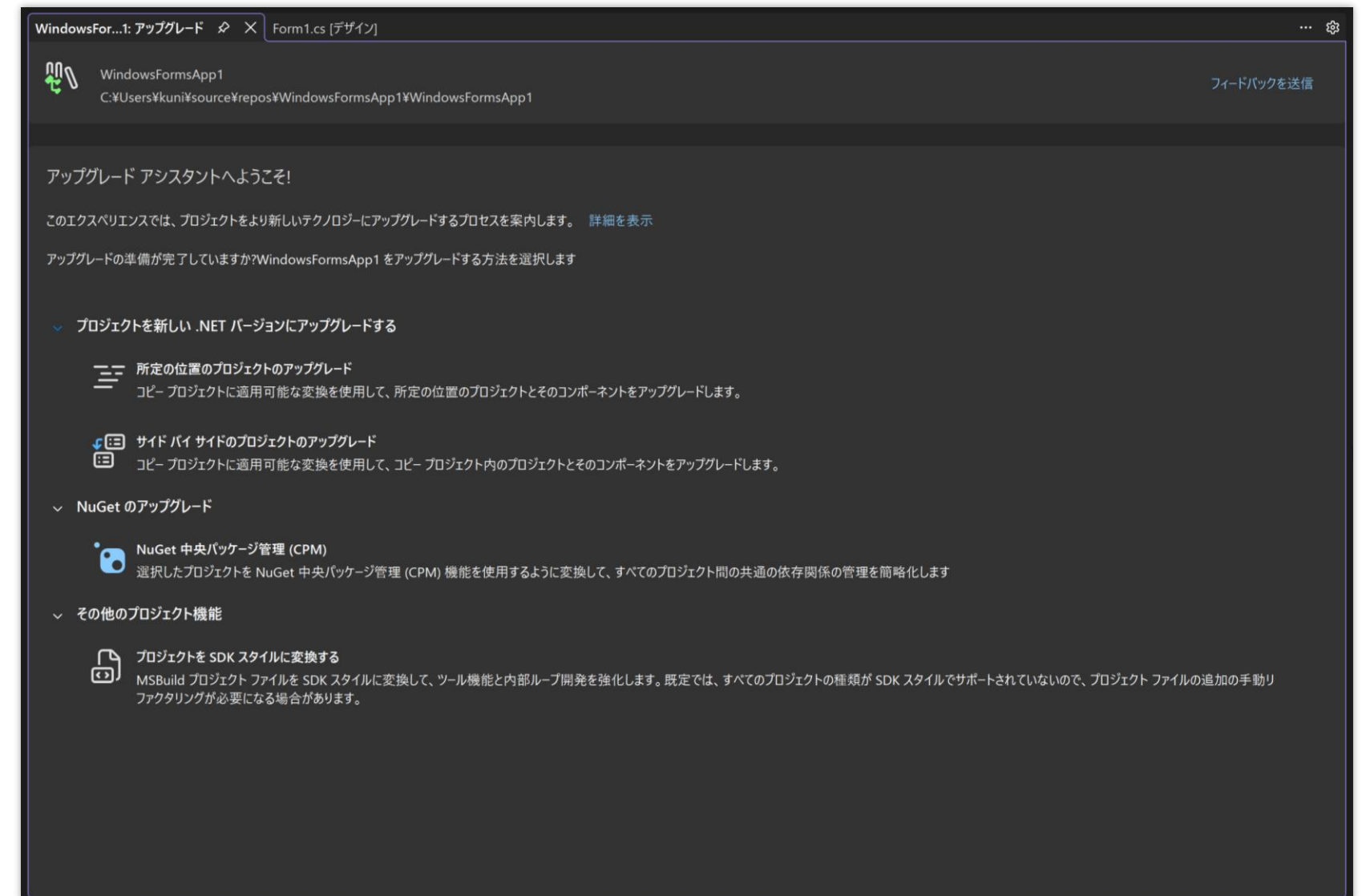
1. .NETアップグレードアシスタントを使用

- .NET Framework → .NET

2. NuGetパッケージを追加 or 更新

- ライセンス、競合エラーを解決

3. 更新履歴で変更点を確認 & 動作を確認



移行方法はヘルプやウェブサイト、ブログで

- 製品ヘルプ、移行ガイド
- 各製品の移行情報
- ナレッジベース
- ブログ記事

The screenshot shows the MESCIUS MultiRow for Windows Forms 12.0J help page. The left sidebar contains a table of contents with the following items:

- MultiRow for Windows Forms 12.0J 移行ガイド
- 旧バージョンとの共存
- 移行ツールの概要
- 移行ツールの操作手順
- 5.0Jからの移行
- 6.0Jからの移行
- 7.0Jからの移行
- 8.0Jからの移行
- 10.0Jからの移行
- 11.0Jからの移行
- Service Pack適用時の移行
- DenpyoMan/El Tabelleからの移行
- .NET Frameworkプロジェクトの.NETヘルプファイルの変更履歴

The main content area displays the following instructions:

ここでは、MultiRowの移行ツールを使用して操作手順を説明します。

1. Visual Studioを移行する必要がある場合には、最初にVisual Studioの変換ウィザードでプロジェクトを移行します。
2. 移行の対象となるプロジェクトがVisual Studioで開かれている場合は、該当プロジェクトを終了してください。
3. スタートメニューの[MESCIUS MultiRow for Windows Forms 12.0J]から「MultiRow for Windows Forms 12.0J 移行ツール」（または<製品インストールフォルダ>\Tools\MRMigratorにある移行ツール（MultiRowMigrationTool.exe）を起動します。
4. 移行ツールを設定します。

The screenshot also shows the MultiRow for Windows Forms 移行ツール (Migration Tool) interface. The tool has the following sections:

- 移行の対象:**
 - 指定したプロジェクトだけを移行する
 - 指定したフォルダ内のプロジェクトをすべて移行する
- 移行する製品構成:**
 - 自動検出: 入力された名前のフォルダが検出されませんでした
 - 上へ
 - 下へ
- オプション:**
 - バックアップを作成する
 - サブフォルダを含める
 - リンクファイルの移行前に確認メッセージを表示する

Buttons at the bottom include 実行 (Execute), 終了 (End), and ヘルプ (Help).

Below the tool interface, there are additional instructions:

[移行の対象] を設定

- 指定したプロジェクトのみ移行する場合は「指定したプロジェクトだけを移行する」を選択します。
- フォルダ内のすべてのプロジェクトを移行する場合は「指定したフォルダ内のプロジェクトをすべて移行する」を選択します。
- 移行するプロジェクトファイルのパスまたはフォルダを指定します。

[移行する製品構成] を設定

- 指定したプロジェクトから自動検出された移行情報が表示されるので、移行情報を確認しチェックボックスをオンにします。

[オプション] を設定

対応環境の最新情報はウェブサイトで

メシウス製品の対応情報をチェック

- .NET 10 対応
- Visual Studio 2026 対応
- Windows 11 対応
- Windows Server 2025 対応

必要システム

[ホーム](#) > [サポート&サービス](#) > [必要システム](#)

Windows 11、Windows Server 2022/2025、Visual Studio 2022/2026への対応状況については、以下のページをご参照ください。

- [Windows 11、Windows Server 2022/2025、Visual Studio 2022/2026対応について](#)

.NET 5~10への対応状況については、以下のページをご参照ください。

- [.NET \(.NET Core\) 対応について](#)

Internet Explorer 11およびMicrosoft EdgeのIEモードは、2022年6月16日でサポートを終了しました。詳細については以下のページをご参照ください。

- [Internet Explorer 11のサポート終了について \(.NET\)](#)
- [Internet Explorer 11のサポート終了について \(JavaScriptライブラリ\)](#)
- [Microsoft Edgeのサポートについて \(.NET\)](#)
- [Microsoft Edgeのサポートについて \(JavaScriptライブラリ\)](#)

<https://developer.mescius.jp/support/system>

最後までご聴講いただき
ありがとうございました

 MESCIOUS