

Mise à jour d'un projet .NET (C#) pour AutoCAD 2025 (.NET 8.0)

À partir de la version 2025, les plugins .NET pour AutoCAD doivent cibler .NET Core 8.0 (ou .NET 8.0) à la place de .NET Framework.

Les plugins ciblant des versions du Framework .NET (4.8 pour AutoCAD 2024) doivent donc être mis à jour pour fonctionner sur AutoCAD 2025 (et les versions ultérieures).

Il s'agit d'une rupture de compatibilité, la deuxième depuis l'intégration de la programmation .NET dans AutoCAD (la première avait eu lieu avec la version 2013 et la division de acad.exe en deux fichiers : acad.exe et accore.dll).

L'objectif de ce document est de montrer comment mettre à jour un projet ciblant le Framework .NET afin qu'il soit compatible avec AutoCAD 2025. Le processus nécessite l'utilisation de Visual Studio 2022 (version 17.8 ou ultérieure) et, bien sûr, d'une version AutoCAD 2025. Comme avec les versions précédentes, il est préférable de référencer les bibliothèques AutoCAD depuis le SDK ObjectARX 2025 téléchargeable depuis [cette page](#).

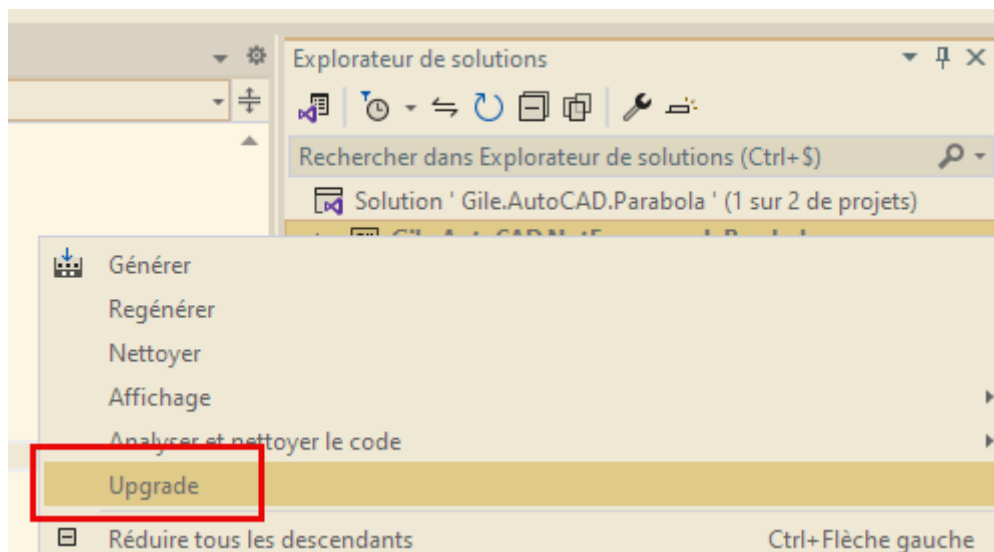
L'assistant de mise à niveau de .NET

Pour faciliter la mise à jour, Microsoft fournit un assistant de mise à niveau. Le plus simple est d'installer l'extension pour Visual Studio (voir [les explications](#)).

La migration depuis Visual Studio

Ouvrir le projet .NET à migrer dans Visual Studio 2022.

Dans l'explorateur de solutions, faire un clic droit sur le projet puis cliquer sur Upgrade.



Choisir : *Upgrade project to a newer .NET version.*

Welcome to the Upgrade Assistant!

This experience will guide you through the process of upgrading your project towards newer technologies. [Learn more](#)

Ready for upgrade? Select how you want to upgrade Gile.AutoCAD.NetFramework.Parabola

- .NET Upgrade project to a newer .NET version**
Upgrade .NET Framework or .NET applications to newer .NET versions, bringing its cross-platform, high-performance capabilities to your product
- Upgrade project features**
Upgrade selected project features without changing whole project and its target framework

Choisir le type de mise à niveau

In-place project upgrade pour remplacer le projet existant

Side-by-side project upgrade pour créer un nouveau projet mis à jour en conservant le projet original

Side-by-side incremental project upgrade pour créer un nouveau projet mis à jour en liant le code du nouveau projet à celui de l'original. Cette option permet de n'avoir à modifier le code que dans un projet, mais rend plus compliquée l'utilisation des nouvelles fonctionnalités de C# 12 qui n'existent pas dans le projet original (C# 7 au plus).

Puis *Next*.

⏪ Upgrade type

What is your upgrade type? Next

- In-place project upgrade**
Upgrades project and its components in place using transformations applicable for the project.
- Side-by-side project upgrade**
Upgrades project and its components in a copy project using transformations applicable for the project.
- Side-by-side incremental project upgrade**
Selects or creates new target project and establishes a link between current and target projects to allow continuous move of project components one at a time.

Choisir *New project*, puis *Next*.

Upgrade type > Upgrade target

What is your upgrade target? Next

- New project**
New project will be created from specified template and associated with current project as an upgrade target
- Existing project**
Existing project of a supported type can be chosen as an upgrade target project

Nommer le nouveau projet, puis *Next*.

Upgrade type > Upgrade target > New project

Upgrading to a new project Next

Project name
Gile.AutoCAD.NetCore.Parabola

Project template
Empty Project

Choisir *.Net 8*, puis *Next*.

Upgrade type > Upgrade target > New project > Target framework

What is your preferred target framework? Next

- .NET 6.0
Supported until novembre, 2024
- .NET 7.0 **STS**
Supported until mai, 2024
- .NET 8.0 LTS**
Supported until novembre, 2026
- .NET 9.0 **PREVIEW**
Try latest preview features
- .NETStandard 2.0
- .NETStandard 2.1

Laisser toutes les cases cochées, puis *Upgrade selection*.



Modification du fichier .csproj

Une fois la migration avec l'assistant terminée, il faut apporter quelques modifications au fichier .csproj. On ouvre ce fichier en double cliquant sur le nouveau projet dans l'explorateur de solution.

Dans la première balise PropertyGroup, remplacer :

```
<TargetFramework>net8.0</TargetFramework>
```

Par :

```
<TargetFramework>net8.0-windows</TargetFramework>
```

Ajouter une balise PlatformTarget :

```
<PlatformTarget>x64</PlatformTarget>
```

Et, pour pouvoir utiliser les [types référence nullable](#) (optionnel), ajouter une balise Nullable:

```
<Nullable>enable</Nullable>
```

Ajouter une balise ItemGroup pour éviter les erreurs de conflit entre .NET Framework et .NET 8.0.

```
<ItemGroup>  
  <FrameworkReference Include="Microsoft.WindowsDesktop.App" />  
</ItemGroup>
```

Modifier les chemins des bibliothèques AutoCAD (accoremgd.dll, acdbmgd.dll, acmgd.dll, ...) pour référencer ceux d'AutoCAD 2025.

Si le fichier contient une balise PropertyGroup contenant des balises StartAction, StartProgram, StartArguments pour démarrer automatiquement AutoCAD en mode Debug, elle peut être supprimée. Avec .NET Core les actions et options de démarrage sont enregistrées dans un fichier json (voir plus bas).

Le fichier .csproj devrait ressembler à ça :

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0-windows</TargetFramework>
    <OutputType>Library</OutputType>
    <GenerateAssemblyInfo>>false</GenerateAssemblyInfo>
    <AssemblyTitle>Gile.AutoCAD.NetCore.Parabola</AssemblyTitle>
    <Company>gileCAD</Company>
    <Product>Gile.AutoCAD.NetCore.Parabola</Product>
    <Copyright>Copyright © gileCAD 2022</Copyright>
    <AssemblyVersion>1.0.0.0</AssemblyVersion>
    <FileVersion>1.0.0.0</FileVersion>
    <PlatformTarget>x64</PlatformTarget>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <FrameworkReference Include="Microsoft.WindowsDesktop.App" />
  </ItemGroup>

  <ItemGroup>
    <Reference Include="AcCoreMgd">
      <HintPath>C:\ObjectARX 2025\inc\AcCoreMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="AcDbMgd">
      <HintPath>C:\ObjectARX 2025\inc\AcDbMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="AcMgd">
      <HintPath>C:\ObjectARX 2025\inc\AcMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
  </ItemGroup>

  <!--[...]-->

  <ItemGroup>
    <None Update="start.scr">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </None>
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.CSharp" Version="4.7.0" />
    <PackageReference Include="System.Data.DataSetExtensions" Version="4.5.0" />
  </ItemGroup>
</Project>
```

Enregistrer les modifications.

Modifier les propriétés du projet

Certaines des modifications apportées au fichier .csproj auraient aussi pu être faites depuis les propriétés du projet.

Pour lancer automatiquement AutoCAD au démarrage en mode Debug, il faut modifier les actions et options de démarrage en créant un « Profil de lancement de débogage ».

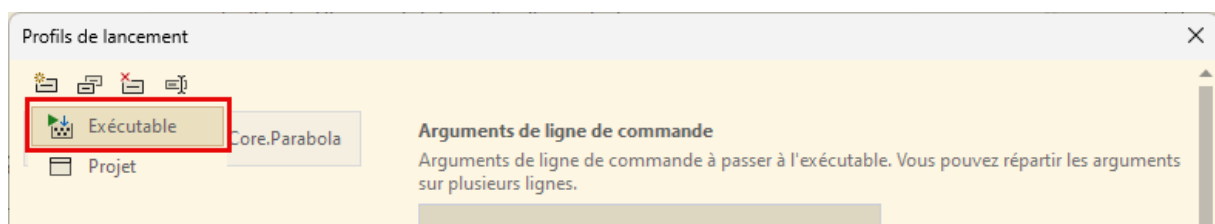
Dans les propriétés du projet, ouvrir la section Déboguer.



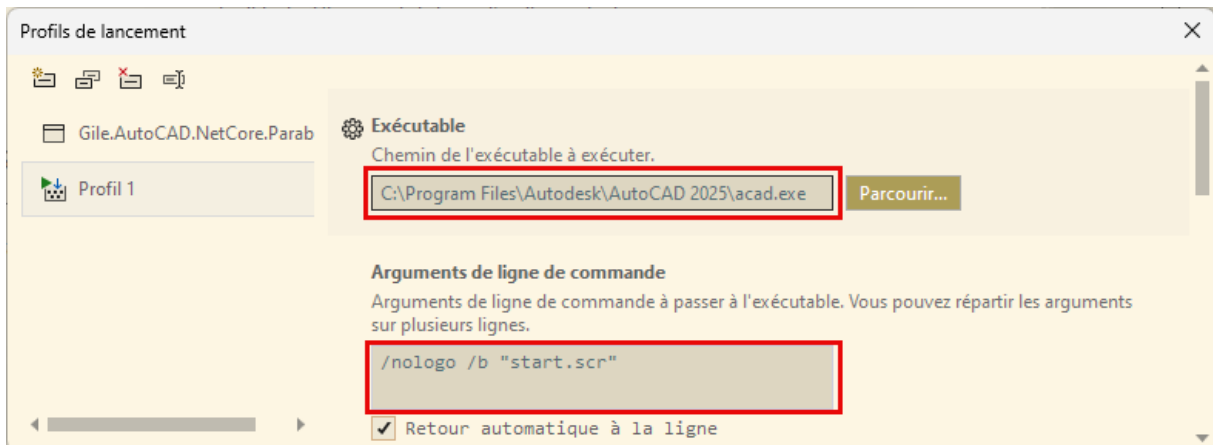
Puis cliquer sur Profils de lancement de débogage pour ouvrir la boîte de dialogue Profils de lancement et cliquer sur Créer un profil.



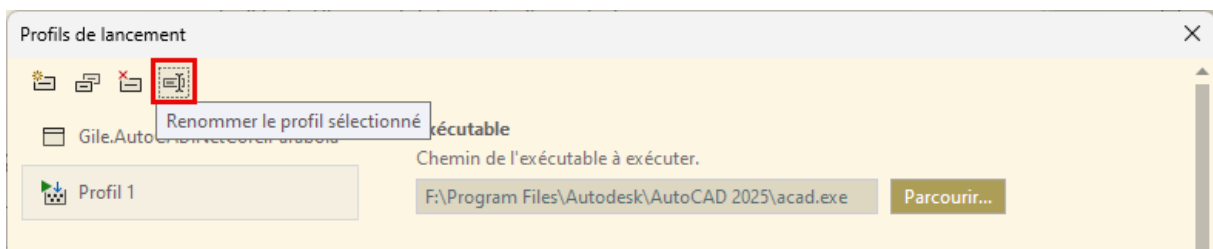
Choisir exécutable.



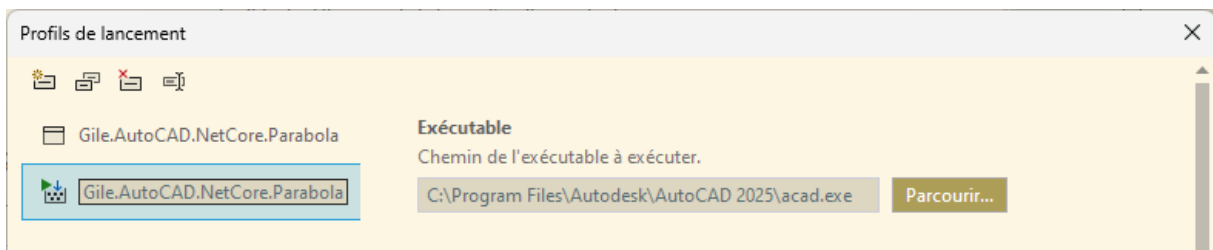
Spécifier le chemin de la version d'AutoCAD à ouvrir et, éventuellement des arguments de ligne de commande, par exemple pour lancer un script au démarrage d'AutoCAD qui chargera le projet.



Puis cliquer sur le bouton Renommer le profil sélectionné.



Et entrer le nom du profil original pour l'écraser.

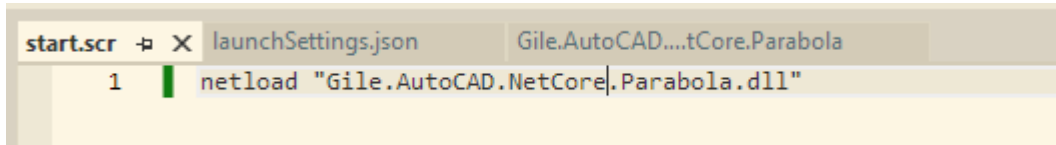


Valider et fermer la boîte de dialogue.

Ces changements ont été enregistrés dans le fichier launchSettings.json du dossier Properties du projet. Il aurait été possible de créer ou modifier ce fichier directement dans Visual Studio.

```
{
  "profiles": {
    "Gile.AutoCAD.NetCore.Parabola": {
      "commandName": "Executable",
      "executablePath": "C:\\Program Files\\Autodesk\\AutoCAD 2025\\acad.exe",
      "commandLineArgs": "/nologo /b \"start.scr\""
    }
  }
}
```

Si dans les arguments de ligne de commande un script est lancé pour charger automatiquement le projet, il faut modifier le nom de la DLL pour qu'il corresponde à celui du nouveau projet.

A screenshot of a script editor window. The title bar shows 'start.scr' and two tabs: 'launchSettings.json' and 'Gile.AutoCAD...tCore.Parabola'. The main area contains a single line of code: '1 netload "Gile.AutoCAD.NetCore.Parabola.dll"'.

```
start.scr  X launchSettings.json  Gile.AutoCAD...tCore.Parabola
1 netload "Gile.AutoCAD.NetCore.Parabola.dll"
```

Modifier le code des fichiers .cs

Lors de la migration, le code des fichiers source a été copié tel quel dans le nouveau projet. Pour corriger les éventuelles erreurs et utiliser les nouvelles fonctionnalités, modifier le contenu des fichiers (la fenêtre Liste d'erreurs et l'intellisense fournissent une aide précieuse).

Il est aussi préférable de changer l'espace de nom pour le faire correspondre à la structure de dossiers (le nom du projet) avec un Rechercher/Remplacer dans le projet existant. Ceci résoudra les éventuels problèmes avec les fichiers automatiquement générés par Visual Studio (fichiers de ressources par exemple).

Définir en tant que projet de démarrage le nouveau projet et lancer le débogage pour tester si tout fonctionne comme espéré.