hopp
migrating complex data.
faster, cheaper, better.

migFx 1.7 – Port from .NET Framework

This guide contains information about how the refactorings related to the port of the common migFx libraries from .NET Framework to netstandard 2.0.

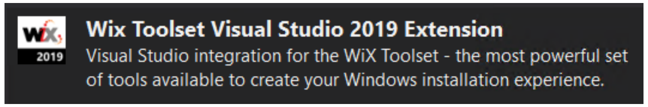
The document also includes the refactorings related to the new implementation of Generation, Debug and Deploy in the migFx Visual Studio extension

Preparation

To work on the refactoring described in this document, your workstation must meet these requirements.

WIX

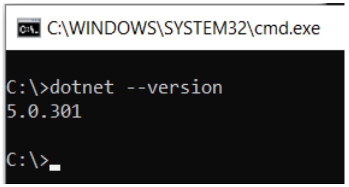
All installer projects in the MigFx Visual Studio solution have now been ported to WIX. Please ensure that you have the WIX Toolset installed (Downloads (wixtoolset.org)) and the WIX Extension added to Visual Studio.



Wix Toolset Visual Studio 2019 Extension
Visual Studio integration for the WiX Toolset - the most powerful set of tools available to create your Windows installation experience.

Net 5.0

Please ensure that minimum Microsoft NET SDK 5.0.301 for Visual Studio is installed (Download .NET 5.0 (Linux, macOS, and Windows) (microsoft.com))



```
C:\>dotnet --version
5.0.301

C:\>
```

Visual Studio Deploy & Debug

As part of the new deploy and debug mechanism via the Director in the MigFx Visual Studio extension manual refactorings must be applied to the MigFx.config files in your Visual Studio solution. In addition, manual updates must be carried out in the Master database.

MigFx.config files

Firstly, all *MigFx.config* files in Visual Studio solutions containing MigFx Engine projects must be renamed to *MigFx.manifest*.

Secondly, the content of the manifest files must be refactored to account for the revised Debug and Deploy mechanism and for the new format of the command line parameter for the code generator.

Below a sample file the showing the file content before refactoring.

```
<?xml version="1.0" encoding="utf-8"?>
<Config>

  <Deploy name="System Test" type="xcopy" deployPath="D:\MigFx\Runtime\Engine\Workshop"/>
  <Deploy name="Integration Test" type="install" deployPath="D:\MigFx\Runtime\Engine\Workshop\Installer" server="."/>

  <ProjectType type="Source" xcopyDeployFolder="Source">
    <PublishedXml>Workshop.SourceEngineCustom\PublishedSource.xml</PublishedXml>
    <Generate>
      <Cmd name ="MigFx Source">
        <Generator>"C:\Program Files (x86)\HoppTech\MigFx.EngineFramework\1.0\Source.EngineGenerator.exe"</Generator>
        <Arguments>/abstract=false /DeveloperMode /verbose /operation:GenerateEngine /projectName:Workshop
/input:"Workshop.SourceEngineCustom\PublishedSource.xml" /output:.</Arguments>
      </Cmd>
    </Generate>
    <Project name="Workshop.SourceEngineCustom" checkOutProjectFile="true" xcopyDeploy="true" buildOrder="1">
      <Folder localPath="_Generated" />
    </Project>
    <Project name="Workshop.SourceEngineGenerated" checkOutProjectFile="true" buildOrder="0">
      <Folder localPath="" />
    </Project>
    <SetupProject>Workshop.SourceSetup</SetupProject>
  </ProjectType>

  <ProjectType type="Target" xcopyDeployFolder="Target">
    <PublishedXml>Workshop.TargetEngine\PublishedTarget.xml</PublishedXml>
    <Generate>
      <Cmd name ="MigFx Target">
        <Generator>"C:\Program Files (x86)\HoppTech\MigFx.EngineFramework\1.0\Target.EngineGenerator.exe"</Generator>
        <Arguments>-V:true -publishedXml:"Workshop.TargetEngine\PublishedTarget.xml" -output:"Workshop.TargetEngine\_Generated"</Arguments>
      </Cmd>
    </Generate>
    <Project name="Workshop.TargetEngine" checkOutProjectFile="false" xcopyDeploy="true">
      <Folder localPath="_Generated" />
    </Project>
    <SetupProject>Workshop.TargetSetup</SetupProject>
  </ProjectType>
</Config>
```

And here the same file after refactoring.

```
<?xml version="1.0" encoding="utf-8"?>
<Manifest>
  <ProjectType type="Source">
    <PublishedXml>Workshop.SourceEngineCustom\PublishedSource.xml</PublishedXml>
    <Generate>
      <Cmd name ="MigFx Source">
        <FileName>SourceEngineGenerator</FileName>
        <Arguments>--verbose --developermode --projectName Workshop --input "Workshop.SourceEngineCustom\PublishedSource.xml" --output .</Arguments>
      </Cmd>
    </Generate>
    <Project name="Workshop.SourceEngineCustom" checkOutProjectFile="true" deploy="true" buildOrder="1">
      <Folder localPath="_Generated" />
    </Project>
    <Project name="Workshop.SourceEngineGenerated" checkOutProjectFile="true" buildOrder="0">
      <Folder localPath="" />
    </Project>
  </ProjectType>

  <ProjectType type="Target">
    <PublishedXml>Workshop.TargetEngine\PublishedTarget.xml</PublishedXml>
    <Generate>
      <Cmd name ="MigFx Target">
        <FileName>TargetEngineGenerator</FileName>
        <Arguments>--verbose --publishedXml "Workshop.TargetEngine\PublishedTarget.xml" --output "Workshop.TargetEngine\_Generated"</Arguments>
      </Cmd>
    </Generate>
    <Project name="Workshop.TargetEngine" checkOutProjectFile="false" deploy="true">
      <Folder localPath="_Generated" />
    </Project>
  </ProjectType>
</Manifest>
```

The list of refactorings to do is:

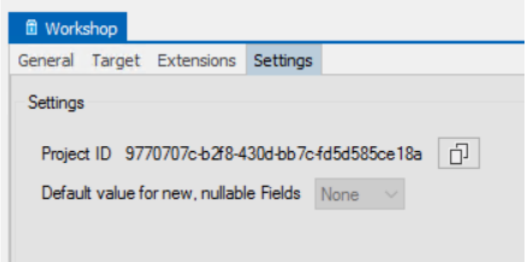
1. Rename root node from *Config* to *Manifest*
2. Remove all *Deploy* nodes
3. Remove the *xcopyDeployFolder* attribute from all *ProjectType* nodes
4. Change the name of all *ProjectType/Generate/Cmd/Generator* nodes to *FileName*
5. For the content of the *FileName* nodes referring the Source- and Target EngineGenerator, remove the preceding full path – leaving only the executeable filename itself (the install of the Engine Framework adds the path to the executables to the PATH environment variable for the machine)
    a. *FileName* should be *TargetEngineGenerator* and *SourceEngineGenerator*, respectively
6. The content of the *ProjectType/Generate/Cmd/Arguments* nodes must be updated to follow the option naming of the new netstandard CommandLineParser using by the ported engine generators
    a. Replace the preceding / character before all options with -- (2 dashes)
    b. Replace any : (colon) or = (equal) characters between option names and values with a space
    c. For the Source Engine Generator
        i. Remove the *operation* option (GenerateEngine)
7. Remove all *xcopyDeploy="true"* attributes from *ProjectType/Project* nodes
8. Add a *deploy="true"* attribute to the *TargetEngine* Project node
9. Add a *deploy="true"* attribute to the *SourceEngineCustom* Project node
10. Remove any *ProjectType/SetupProject* nodes

Master Database

When deploying through the Director, the VsExtension will read the id of the Studio Map project from the header/projectId element in the published xml file and use this to identify the engine in the Master Database.

[mig_director].[Engine]

As a consequence, the EngineID in the [mig_director].[Engine] table must be manually updated to contain the actual project id of the corresponding map in Studio. The id can be found either in the published xml file (as mentioned above) or in Studio on the Settings tab of the root node in the Project Explorer View



Settings

Project ID 9770707c-b2f8-430d-bb7c-fd5d585ce18a

Default value for new, nullable Fields None

[mig_director].[Project]

The [mig_director].[Project] table must be manually updated and the SourceEngineID and TargetEngineID columns updated to the corresponding, new values in the [mig_director].[Engine] table

Installation

MigFx version 1.7 no longer installs the public interface libraries in GAC. Instead, the libraries are installed in a local folder, from where they must be references by Souce- and Target Engine projects as well as any on-premises MigFx extensions.

By default, the libraries are installed to *C:\Program Files\HoppTech\migFx\Libraries.*

Only install the MigFx.Libraries on workstations where Visual Studio is used for MigFx Engine generation and deployment and/or MigFx Extension development.

OBS: It is not necessary to install the MigFx.Libraries on Master or Migration servers.

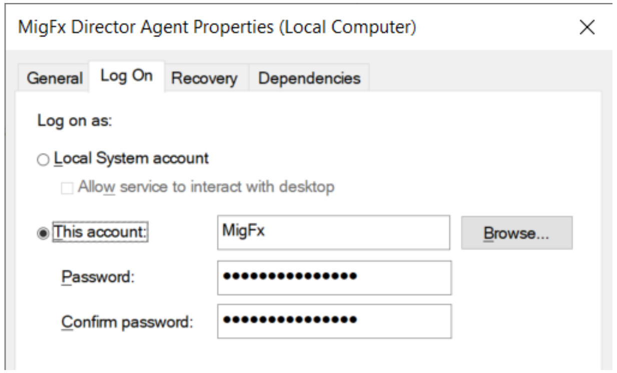
To ensure a clean install, it is recommended to

1. Stop all Tracks
2. Stop the Director Agent Service
3. Stop all MigFx app pools in IIS
4. Uninstall the MigFx Visual Studio Extension

5. Close all instances of Visual Studio
6. Close all instances of migFx Studio and migFx Director
7. Uninstall all MigFx components and on-premises extensions including Tracker, Director Client and AuthService
8. Ensure that all MigFx libraries are removed from GAC and manually delete any that are not
9. Install the MigFx.Libraries and perform all refactoring as described below
10. Reinstall all MigFx components and on-premises extensions

Installing the Director Agent service

The installer no longer prompts for the credentials for the Director Agent. Instead, set the credentials after install using the Services.msc admin console.



Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: MigFx

Password: ••••••••••••••••

Confirm password: ••••••••••••••••

Because of the port to Net 5.0, the config file for the Agent Service in the install folder has changed to *MigFx.Agent.dll.config*. Please update this file as appropriate.

Web.config files

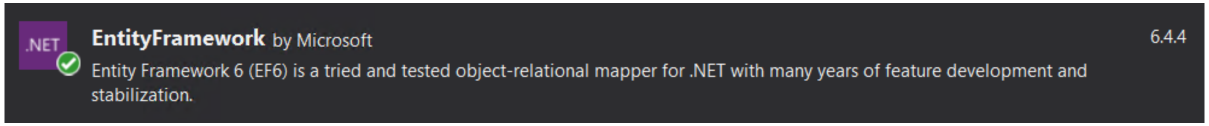
The newly installed web config files for Tracker and the Director web services contains important assembly redirection information. If you have any saved copies of web config files, please update the new config files from your copies instead of replacing them.

Engine projects

Firstly, if you are using Nuget in any projects, it is recommended to migrate from the (obsolete) packages.config file to the new PackageReference element in the csproj file. This is easily done by right-clicking the packages.config files in Visual Studio and choose *Migrate packages.config to PackageReference*. Be aware that the migration tool stores migration an html report in subfolder in the solution root. You may want to clean that up before committing to Source Control.

You may find it easiest to manually edit the csproj files to exchange the old MigFx references with the new ones.

For engine projects that reference the EntityFramework (most should), you can choose either to reference the 2 EntityFramework libraries installed with MigFx.Libraries. However, if possible, it is recommended that you use the EntityFramework Nuget package instead:



EntityFramework by Microsoft

6.4.4

Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature development and stabilization.

Or, as PackageReference in the csproj file:

```
<PackageReference Include="EntityFramework">
      <Version>6.4.4</Version>
</PackageReference>
```

Important

All projects must be updated to minimum target .NET Framework 4.7.2. Any project targeting earlier .NET Framework versions will not be able to reference the new netstandard 2.0 MigFx Libraries.

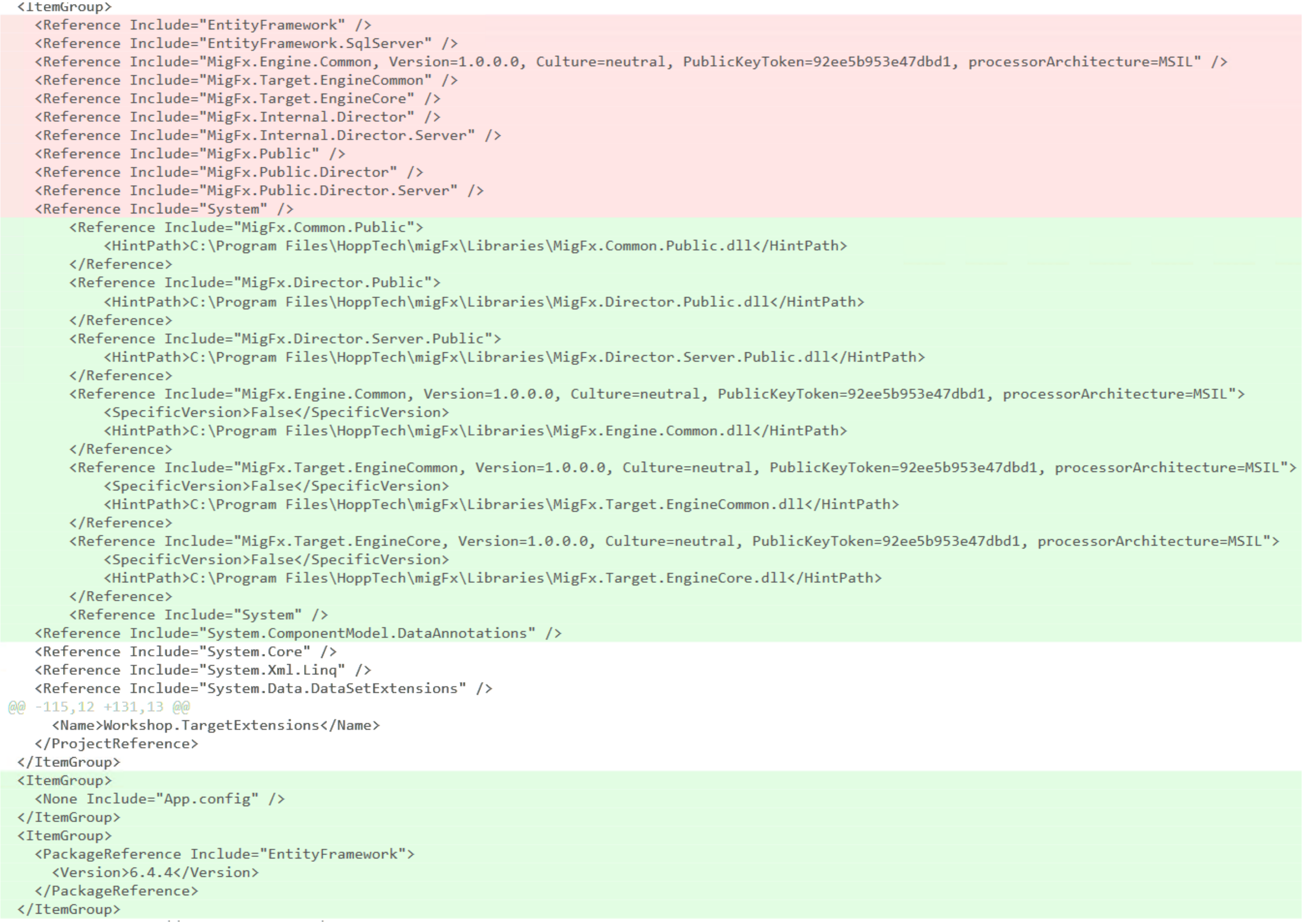
Update project references

Update all references in the Visual Studio engine projects to reference the migFx.Libraries. You can update in Visual Studio Solution Explorer by deleting the old references and re-adding the correct references to the new libraris.

Or you can manually edit the csproj files as illustrated in the following sections.

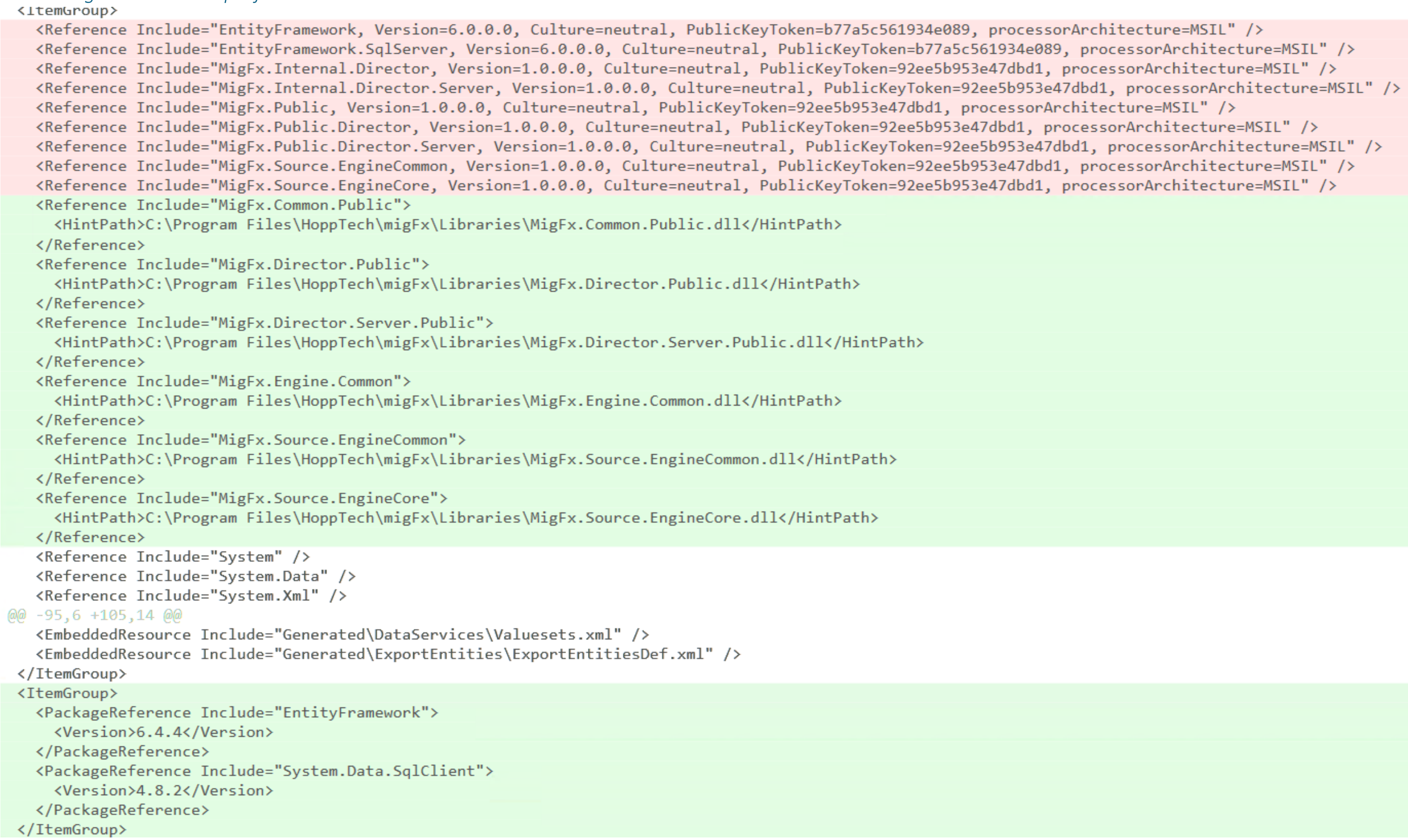
For the Microsoft EntityFramework, you can reference the Nuget package provided by Microsoft, or you can reference the EntityFramework libraries provided with the migFx Libraries.

Target Engine projects



```
  <ItemGroup>
    <Reference Include="EntityFramework" />
    <Reference Include="EntityFramework.SqlServer" />
    <Reference Include="MigFx.Engine.Common, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
    <Reference Include="MigFx.Target.EngineCommon" />
    <Reference Include="MigFx.Target.EngineCore" />
    <Reference Include="MigFx.Internal.Director" />
    <Reference Include="MigFx.Internal.Director.Server" />
    <Reference Include="MigFx.Public" />
    <Reference Include="MigFx.Public.Director" />
    <Reference Include="MigFx.Public.Director.Server" />
    <Reference Include="System" />
        <Reference Include="MigFx.Common.Public">
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Common.Public.dll</HintPath>
        </Reference>
        <Reference Include="MigFx.Director.Public">
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Director.Public.dll</HintPath>
        </Reference>
        <Reference Include="MigFx.Director.Server.Public">
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Director.Server.Public.dll</HintPath>
        </Reference>
        <Reference Include="MigFx.Engine.Common, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL">
            <SpecificVersion>False</SpecificVersion>
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Engine.Common.dll</HintPath>
        </Reference>
        <Reference Include="MigFx.Target.EngineCommon, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL">
            <SpecificVersion>False</SpecificVersion>
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Target.EngineCommon.dll</HintPath>
        </Reference>
        <Reference Include="MigFx.Target.EngineCore, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL">
            <SpecificVersion>False</SpecificVersion>
            <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Target.EngineCore.dll</HintPath>
        </Reference>
        <Reference Include="System" />
    <Reference Include="System.ComponentModel.DataAnnotations" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
@@ -115,12 +131,13 @@
      <Name>Workshop.TargetExtensions</Name>
    </ProjectReference>
  </ItemGroup>
  <ItemGroup>
    <None Include="App.config" />
  </ItemGroup>
  <ItemGroup>
    <PackageReference Include="EntityFramework">
      <Version>6.4.4</Version>
    </PackageReference>
  </ItemGroup>
```

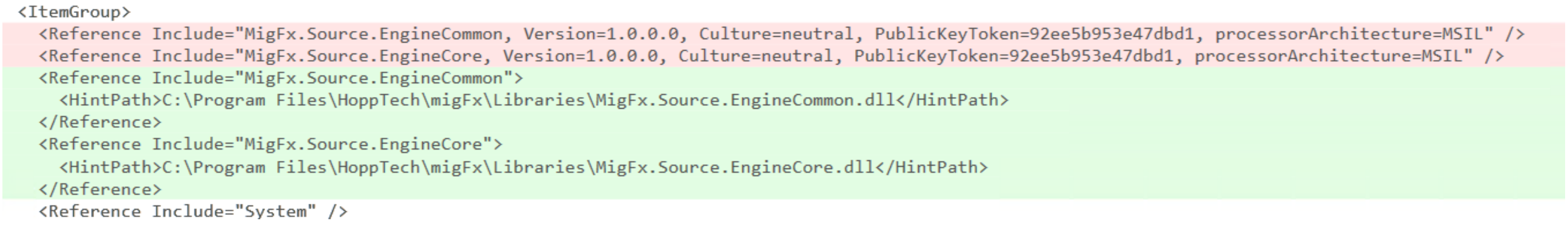
SourceEngineGenerated projects



```
 <ItemGroup>
   <Reference Include="EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL" />
   <Reference Include="EntityFramework.SqlServer, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Internal.Director, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Internal.Director.Server, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Public, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Public.Director, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Public.Director.Server, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Source.EngineCommon, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Source.EngineCore, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
   <Reference Include="MigFx.Common.Public">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Common.Public.dll</HintPath>
   </Reference>
   <Reference Include="MigFx.Director.Public">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Director.Public.dll</HintPath>
   </Reference>
   <Reference Include="MigFx.Director.Server.Public">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Director.Server.Public.dll</HintPath>
   </Reference>
   <Reference Include="MigFx.Engine.Common">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Engine.Common.dll</HintPath>
   </Reference>
   <Reference Include="MigFx.Source.EngineCommon">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Source.EngineCommon.dll</HintPath>
   </Reference>
   <Reference Include="MigFx.Source.EngineCore">
     <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Source.EngineCore.dll</HintPath>
   </Reference>
   <Reference Include="System" />
   <Reference Include="System.Data" />
   <Reference Include="System.Xml" />
@@ -95,6 +105,14 @@
   <EmbeddedResource Include="Generated\DataServices\Valuesets.xml" />
   <EmbeddedResource Include="Generated\ExportEntities\ExportEntitiesDef.xml" />
 </ItemGroup>
 <ItemGroup>
   <PackageReference Include="EntityFramework">
     <Version>6.4.4</Version>
   </PackageReference>
   <PackageReference Include="System.Data.SqlClient">
     <Version>4.8.2</Version>
   </PackageReference>
 </ItemGroup>
```

OBS: You also need to add Nuget package System.Data.SqlClient

SourceEngineCustom projects



```
<ItemGroup>
  <Reference Include="MigFx.Source.EngineCommon, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
  <Reference Include="MigFx.Source.EngineCore, Version=1.0.0.0, Culture=neutral, PublicKeyToken=92ee5b953e47dbd1, processorArchitecture=MSIL" />
  <Reference Include="MigFx.Source.EngineCommon">
    <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Source.EngineCommon.dll</HintPath>
  </Reference>
  <Reference Include="MigFx.Source.EngineCore">
    <HintPath>C:\Program Files\HoppTech\migFx\Libraries\MigFx.Source.EngineCore.dll</HintPath>
  </Reference>
  <Reference Include="System" />
```

Utility

The MigFx.Utility command line tool is used for validation of Translation Valuesets entries in the Tracker. The tools extracts the Translation Valuesets to a file that can be used for external validation and, as a second step, the tool can read a file with validation results and mark the Translation Valueset entries in the Tracker with any validation errors.

Due to the change of the Command Line Parser used by MigFx, the format of the MigFx.Utility command line has changed. Please update all calls to MigFx.Utility accordingly.

Previous command line format:

```
MigFx.Utility translation -action=export -trackID=(track Guid) -path=(foldername) -file=(filename)
MigFx.Utility translation -action=import -trackID=(track Guid) -path=(foldername) -file=(filename)
MigFx.Utility translation -action=backup -trackID=(track Guid)
```

New command line format:

```
MigFx.Utility translation --action export --trackID (track Guid) --path (foldername) --file (filename)
MigFx.Utility translation --action import --trackID (track Guid) --path (foldername) --file (filename)
MigFx.Utility translation --action backup --trackID (track Guid)
```

Changes

- Option names must be prefixed with -- (2 dashes)
- Replace any = (equal sign) or : (colon) between option name and option value with a single space

Extension projects

Please update any extension projects, for instance custom unloaders and dataservices implementations, to target minimum .NET Framework 4.7.2 and to reference the new migFx Libraries as outlined above.

If you have implemented any DataServices extensions that plug into the Studio and Director Runtime, you need to port your installer from the current Visual Studio Installer vdproject type to a WIX installer project. This includes porting the install custom action as well.

Please contact hopp for assistance to port DataServices extension installers.

Excel Utilities

The *MigExcelReader* and *MigExcelWriter* utility classes have moved.

They used to be in the *MigFx.DataServices.Public.dll* library and the *MigFx.DataServices* namespace.

They have moved to a new *MigFx.ExcelUtilities.Public.dll* library and the *MigFx.Excel* namespace.

Please update any extensions using the MigFx Excel utility classes accordingly.