How To: Add API Composites to the Ed-Fi ODS / API Solution

This section outlines the steps necessary to integrate and activate the Ed-Fi Composite definitions for use in an Ed-Fi ODS / API. It is assumed that the Ed-Fi ODS / API has been successfully downloaded and is running in a local environment per the instructions in the Getting Started documentation.

To add Composites to an Ed-Fi ODS / API, you can use the recommended approach of installing the Ed-Fi API Composite project template for Visual Studio, or you can perform the corresponding steps manually. The instructions to use Visual Studio template follow, and the manual steps are listed in Appendix A of this page.

The steps in Visual Studio can be summarized as:

The steps can be summarized as:

- Step 1. Create the Composites Project
- Step 2. Integrate Composites into the Solution
- Step 3. Run Code Generation and Verify Changes

Each step is outlined in detail, below.

Downloads

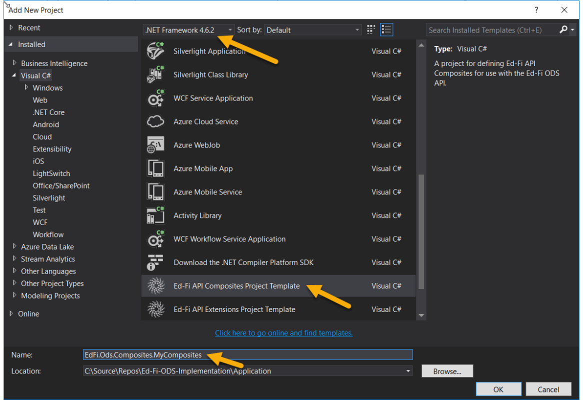
The following GitHub link contains source files for the Composite described in this article:

Composite Source Files

Step 1. Create the Composites Project

Step 1.1. Add a Composite Project Using the Visual Studio Project Template. Visual Studio Project Template can be installed by following steps in Getting Started - Project Templates Installation section of this documentation.

- To add a project to your Ed-Fi-Ods Visual Studio Solution, **right-click** on the "Composites" folder. Select **Add** > **New Project**.
- In the "Add New Project" dialog, find the "Ed-Fi API Composites Project Template" entry at the bottom of the Visual C# section, as shown below. Make sure you choose Microsoft .NET Framework 4.6.2 or above.
- Enter the project name for the new project and click **OK**. The suggested naming convention for this type of project is something like **EdFi.Ods.Composites.MyComposites**.


Add New Project
?
Recent
Installed
Business Intelligence
Visual C#
Windows
Web
.NET Core
Android
Cloud
Extensibility
iOS
LightSwitch
Office/SharePoint
Silverlight
Test
WCF
Workflow
Azure Data Lake
Stream Analytics
Other Languages
Other Project Types
Modeling Projects
Online
Sort by:
Search Installed Templates (Ctrl+E)
Silverlight Application Visual C#
Silverlight Class Library Visual C#
WCF Service Application Visual C#
Azure Cloud Service Visual C#
Azure WebJob Visual C#
Azure Mobile App Visual C#
Azure Mobile Service Visual C#
Activity Library Visual C#
WCF Workflow Service Application Visual C#
Download the .NET Compiler Platform SDK Visual C#
Ed-Fi API Composites Project Template Visual C#
Ed-Fi API Extensions Project Template Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel

.NET Framework 4.6.2 | Sort by: Default

Type: Visual C#

A project for defining Ed-Fi API Composites for use with the Ed-Fi ODS API.

Name: EdFi.Ods.Composites.MyComposites

Location: C:\Source\Repos\Ed-Fi-ODS-Implementation\Application

Step 1.2. Update the **Marker Interface** file.

To integrate the Composite with the API, start by ensuring you have a marker interface in the root of your Composites project. in the root of your Composites project.

This interface is an idiom used in the Ed-Fi Visual Studio Solution to enable a strongly typed mechanism for obtaining a reference to the .NET assembly. If you used the Visual Studio Project template to create your composite, a file will already exist — but you'll need to rename the interface and the file to match the convention (e.g., **Marker_EdFi_Ods_Composites_MyComposites.cs**). The marker interface file should have the following code:

```
namespace EdFi.Ods.Composites.MyComposites
{
    public interface Marker_EdFi_Ods_Composites_MyComposites { }
}
```

Step 1.3. Update the **Composites.xml** file to add the appropriate composite definition.

The Visual Studio Project Template creates a sample **Composites.xml** file. Consult API Composite Resources for guidance. The Composite Definition should look something like the following:

```
<?xml version="1.0" encoding="utf-8"?>
<CompositeMetadata organizationCode="ed-fi">
  <Category displayName="My Sample Composites" name="MyComposite">
    <Routes>
      <Route relativeRouteTemplate="/sections/{Section.Id}/
{compositeName}" />
    </Routes>
    <Composites>
      <Composite name="Student">
        <Specification>
          <Parameter name="Section.Id" filterPath="
StudentSectionAssociation->Section.Id" />
        </Specification>
        <BaseResource name="Student">
          <Property name="StudentUniqueId" />
          <Property name="FirstName" />
          <Property name="LastSurname" />
          <Property name="BirthDate" />
        </BaseResource>
      </Composite>
    </Composites>
  </Category>
</CompositeMetadata>
```

Composites cannot include resource model members added via Ed-Fi Extensions. ODS / API composite definitions currently support only the resource models from the Ed-Fi Core.

The CompositeMetadata attribute 'organizationCode' is a required attribute and indicates the 'Organization' the composite belongs to. This value is carried from the xml definition all the way into the API route definitions. The organizationCode in combination with the Category name are used to identify the composite being requested from the API.

Step 1.4. Save the Project.

Step 2. Integrate Composites into the Solution

To integrate the Composite Resources into the solution, perform the following tasks in the **EdFi.Ods.WebApi** project (located in the "Entry Points" folder):

Step 2.1. Add a reference to the new Composites project you constructed in the previous step.

- Open the **EdFi.Ods.WebApi.Startup.ConfigurationSpecificSandboxStartup.cs** file and add the following code in the following steps.
- Add a line to include MyComposites assembly:

EdFi.Ods.WebApi.Startup.ConfigurationSpecificSandboxStartup.cs

```
using EdFi.Ods.Composites.MyComposites;
```

Step 2.2. Locate the `EnsureAssembliesLoaded` method in the same file and add a line for the new extension assembly marker to ensure it is loaded at runtime. The line should read something like the following:

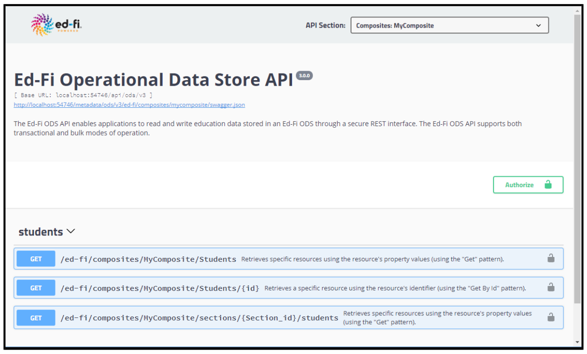
EdFi.Ods.WebApi.Startup.ConfigurationSpecificSandboxStartup.cs

```
 AssemblyLoader.EnsureLoaded<Marker_EdFi_Ods_Composites_MyComposites>();
```

Step 3. Run Code Generation and Verify Changes

Save all modified files, close the Ed-Fi-ODS Visual Studio Solution, then re-run the code generation steps outlined in the Getting Started Guide (i.e., from a PowerShell prompt, run `Initialize-PowershellForDevelopment.ps` script, followed by the `initdev` command). Then run the application and view the Ed-Fi ODS / API using Swagger. The following new API Composite resource should be available:

> ⓘ You may have to clear your browser cache in order for Swagger to display the new resource.



API Section: Composites: MyComposite

Ed-Fi Operational Data Store API 3.0.0

[Base URL: localhost:54746/api/ods/v3]

http://localhost:54746/metadata/ods/v3/ed-fi/composites/mycomposite/swagger.json

The Ed-Fi ODS API enables applications to read and write education data stored in an Ed-Fi ODS through a secure REST interface. The Ed-Fi ODS API supports both transactional and bulk modes of operation.

students

GET `/ed-fi/composites/MyComposite/Students` Retrieves specific resources using the resource's property values (using the "Get" pattern).

GET `/ed-fi/composites/MyComposite/Students/{id}` Retrieves a specific resource using the resource's identifier (using the "Get By Id" pattern).

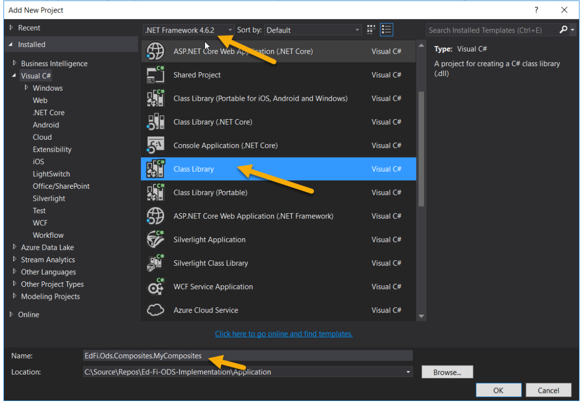
GET `/ed-fi/composites/MyComposite/sections/{Section_id}/students` Retrieves specific resources using the resource's property values (using the "Get" pattern).

The "organizationCode" (ed-fi) mentioned in Step 1.3 can be seen in the Composite URLs above. To successfully retrieve a composite this value (defined in the XML definition) must be present as the first segment of a composite URL.

Appendix A: Adding a Composite Project Manually

Alternatively, you can perform Step 1 manually without using the project template. To add a Composite project to the Ed-Fi ODS / API solution manually, perform the following steps:

1. To add a project to your Ed-Fi-Ods Visual Studio Solution, **right-click** on the "Composites" folder. Select **Add** > **New Project**.
2. In the "Add New Project" dialog, find the "Class Library" entry. Make sure you choose Microsoft .NET Framework 4.6.2 or above.
3. Enter the project name for the new project and click **OK**. The suggested naming convention for this type of project is something like **EdFi.Ods.Composites.MyComposites**.



Add New Project

.NET Framework 4.6.2 Sort by: Default

Type: Visual C#

A project for creating a C# class library (.dll)

Name: EdFi.Ods.Composites.MyComposites

Location: C:\Source\Repos\Ed-Fi-ODS-Implementation\Application

Recent
Installed
Business Intelligence
Visual C#
Windows
Web
.NET Core
Android
Cloud
Extensibility
iOS
LightSwitch
Office/SharePoint
Silverlight
Test
WCF
Workflow
Azure Data Lake
Stream Analytics
Other Languages
Other Project Types
Modeling Projects
Online
ASP.NET Core Web Application (.NET Core) Visual C#
Shared Project Visual C#
Class Library (Portable for iOS, Android and Windows) Visual C#
Class Library (.NET Core) Visual C#
Console Application (.NET Core) Visual C#
Class Library Visual C#
Class Library (Portable) Visual C#
ASP.NET Core Web Application (.NET Framework) Visual C#
Silverlight Application Visual C#
Silverlight Class Library Visual C#
WCF Service Application Visual C#
Azure Cloud Service Visual C#
Search Installed Templates (Ctrl+E)
Click here to go online and find templates.
Browse...
OK
Cancel

4. Create a "marker" interface file in the root of your Composites project named **Marker_EdFi_Ods_Composites_MyComposites.cs** (named to match your assembly). Alternately, you can modify the "Class1.cs" file created in Step 1.3, above. This interface is an idiom used in the Ed-Fi Visual Studio Solution to enable a strongly typed mechanism for obtaining a reference to the .NET assembly. The marker interface file should contain the following code:

```
namespace EdFi.Ods.Composites.MyComposites
{
    public interface Marker_EdFi_Ods_Composites_MyComposites { }
}
```

5. Add an XML file named **Composites.xml** to the root of the project, then add the appropriate composite definition.

   The Composite Definition should look something like the following:

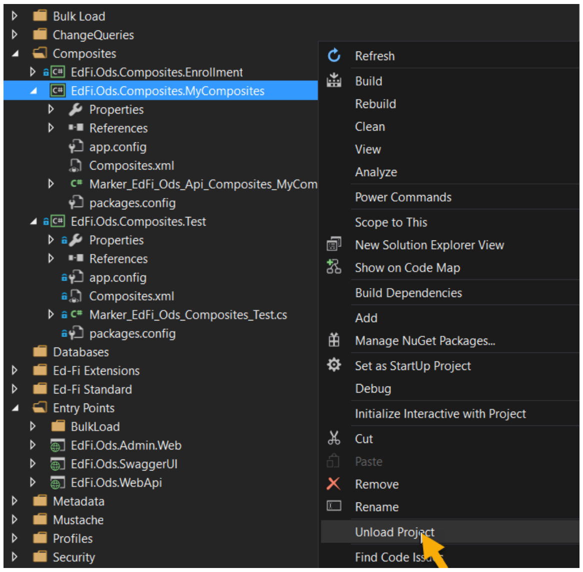
```
<?xml version="1.0" encoding="utf-8"?>
<CompositeMetadata organizationCode="ed-fi">
  <Category displayName="My Sample Composites" name="MyComposite">
    <Routes>
      <Route relativeRouteTemplate="/sections/{Section.Id}/
{compositeName}" />
    </Routes>
    <Composites>
      <Composite name="Student">
        <Specification>
          <Parameter name="Section.Id" filterPath="
StudentSectionAssociation->Section.Id" />
        </Specification>
        <BaseResource name="Student">
          <Property name="StudentUniqueId" />
          <Property name="FirstName" />
          <Property name="LastSurname" />
          <Property name="BirthDate" />
        </BaseResource>
      </Composite>
    </Composites>
  </Category>
</CompositeMetadata>
```

   Composites cannot include resource model members added via Ed-Fi Extensions. Currently only resource models included in the Ed-Fi Core are supported in Composite Definitions.

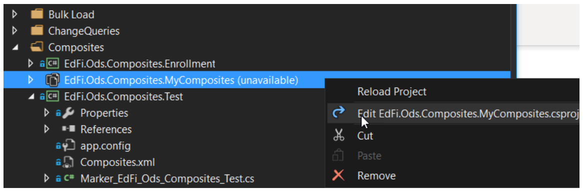
   The CompositeMetadata attribute "organizationCode" is a required attribute and indicates the "Organization" to which the composite belongs. This value is carried from the XML definition all the way into the API route definitions. The organizationCode in combination with the Category name are used to identify the Composite being requested from the API.

6. In the Composites.xml Properties, change the "Build Action" setting to "Embedded Resource".

7. Add the build task section to .csproj file.
   Right click on the created project and choose Unload Project.



8. Edit the project file by right clicking on the project then choose Edit option.


Bulk Load
ChangeQueries
Composites
EdFi.Ods.Composites.Enrollment
EdFi.Ods.Composites.MyComposites (unavailable)
EdFi.Ods.Composites.Test
Properties
References
app.config
Composites.xml
Marker_EdFi_Ods_Composites_Test.cs
Reload Project
Edit EdFi.Ods.Composites.MyComposites.csproj
Cut
Paste
Remove

The .csproj will be opened on the editor now. Add the following build task section text and save.

```
  <UsingTask TaskName="EdFi.Ods.CodeGen.BuildTasks.
CompositeValidationTask" AssemblyFile="$(SolutionDir)..\..\Ed-Fi-
ODS\Application\libs.codegen\EdFi.Ods.CodeGen.dll" />
  <Target Name="AfterBuild">
    <Copy SourceFiles="@(myLibraries)"
DestinationFolder="$(ttLibsFolder)" SkipUnchangedFiles="true" />
    <CompositeValidationTask CompositeDefinitionPath="$(ProjectDir)
Composites.xml" />
  </Target>
```

After saving it, right click on the project and reload it.

9. Save the Project.