Developer Studio Guide

1.0.0

Copyright © 2026 OneStream Software LLC. All rights reserved.

All trademarks, logos, and brand names used on this website are the property of their respective owners. This document and its contents are the exclusive property of OneStream Software LLC and are protected under international intellectual property laws. Any reproduction, modification, distribution or public display of this documentation, in whole or part, without written prior consent from OneStream Software LLC is strictly prohibited.

Table of Contents

Overview

Developer Studio streamlines the custom development process by acting as a stand-alone application that connects with OneStream environments and creates local copies of business rules and workspace assembly files. It enables developers to sync changes between what you have locally and what you have in OneStream. Developers can also leverage IntelliSense within their Integrated Development Environment (IDE) of choice to accelerate the development process.

With Developer Studio, you can:

- Set up and manage repositories for project files.
- Push and pull changes to files between the OneStream server and your local environment.
- Modify code locally in any IDE.
- Download reference assemblies and integrate IntelliSense capabilities.

Setup and Installation

Before you install Developer Studio, review these details.

Dependencies

Component	Description
9.2.0	Minimum platform version required to use Developer Studio

Considerations

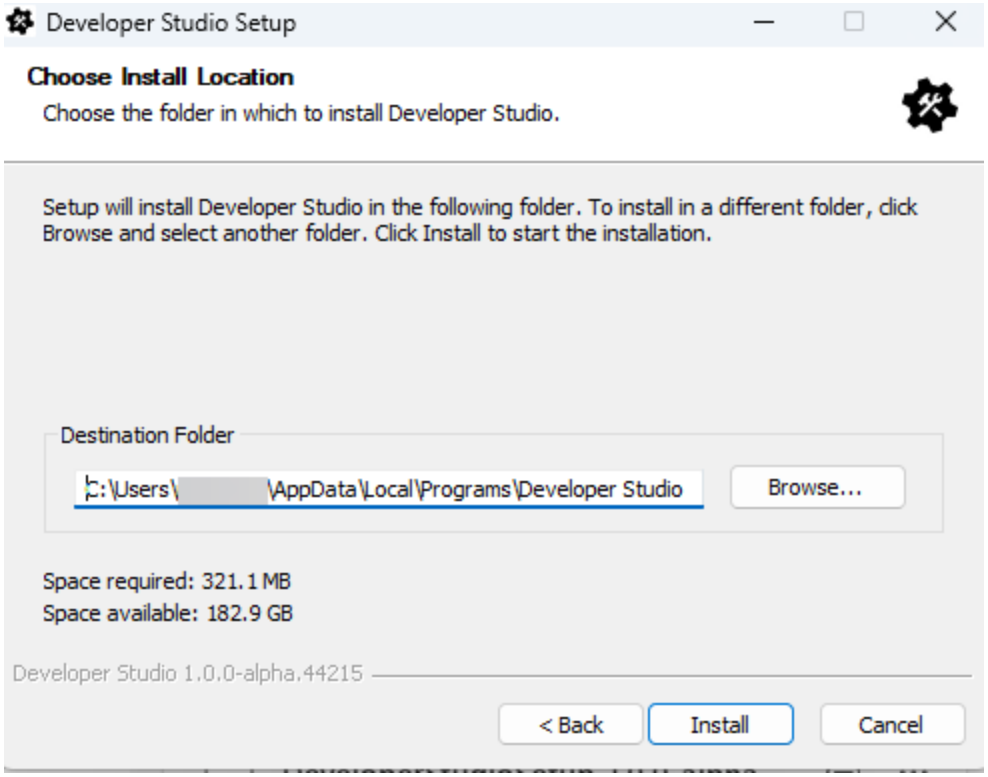
	Installer Deployment
Operating system	Windows
Type of install	Traditional installation. The Developer Studio application displays in the Start menu and Add or Remove programs.
End-user deployment	OneStream Solution Exchange
Requires local administrator	No
Version upgrade	Manual upgrade with installer file

	Installer Deployment
Use multiple versions on the same machine	No
Connect to different servers from single installed instance	Yes

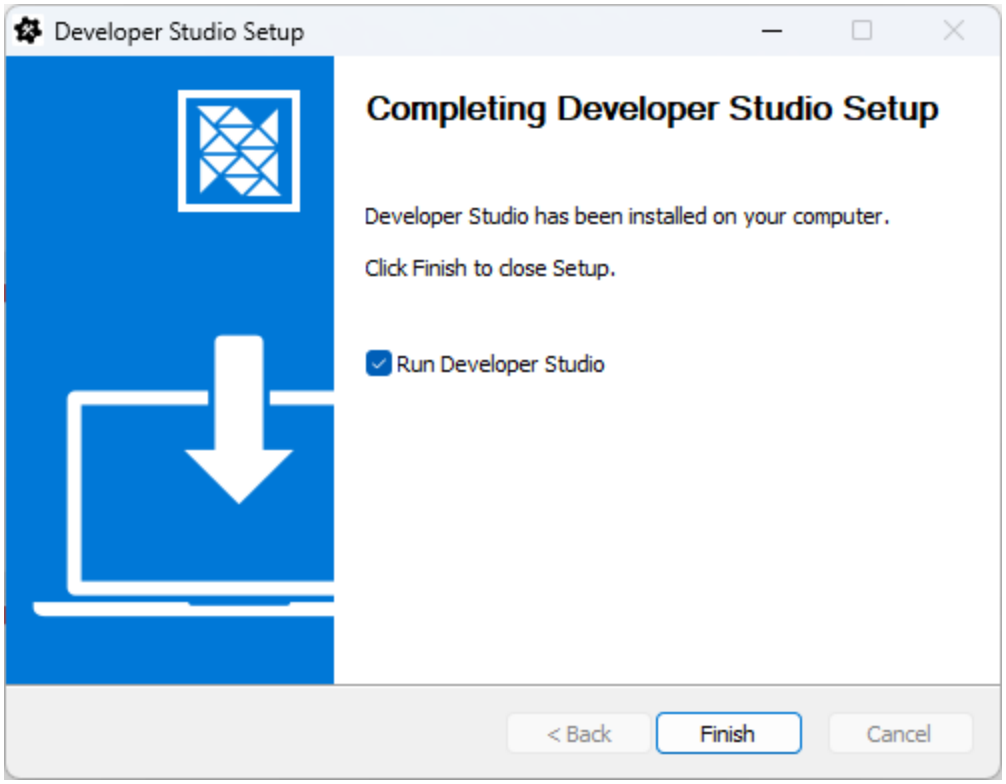
Install Developer Studio

The Developer Studio installer is available on Solution Exchange. Complete the following steps to install Developer Studio:

1. Double-click the Developer Studio installer file and then click the **Next** button.
2. If necessary, change the folder path. Click the **Install** button.

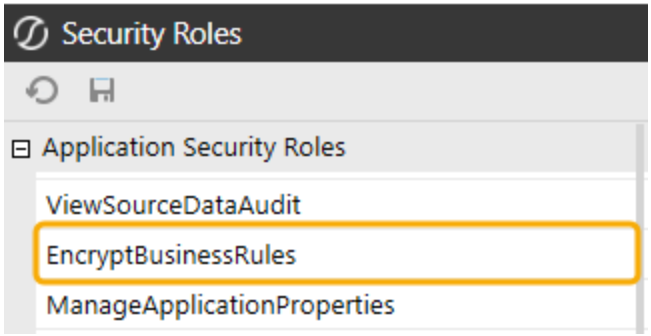


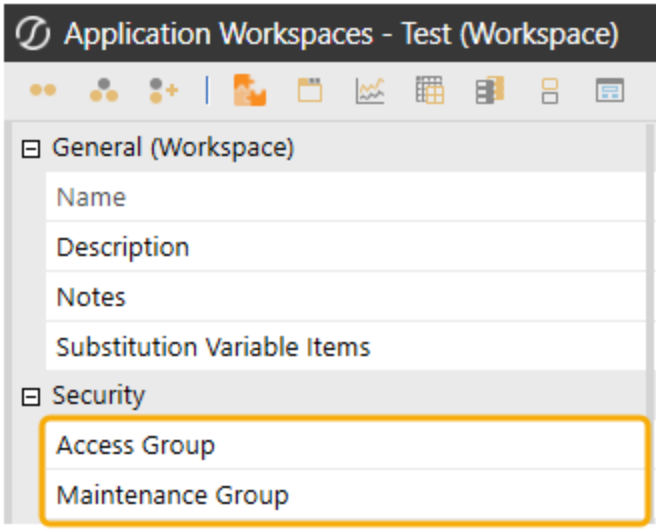
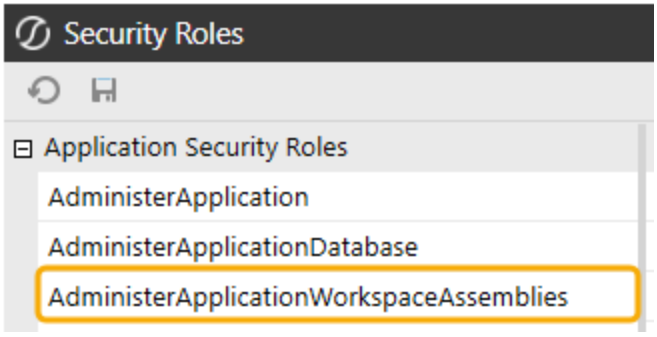
3. When the progress bar for the installation completes, click the **Next** button.
4. Click the **Finish** button to complete the installation.

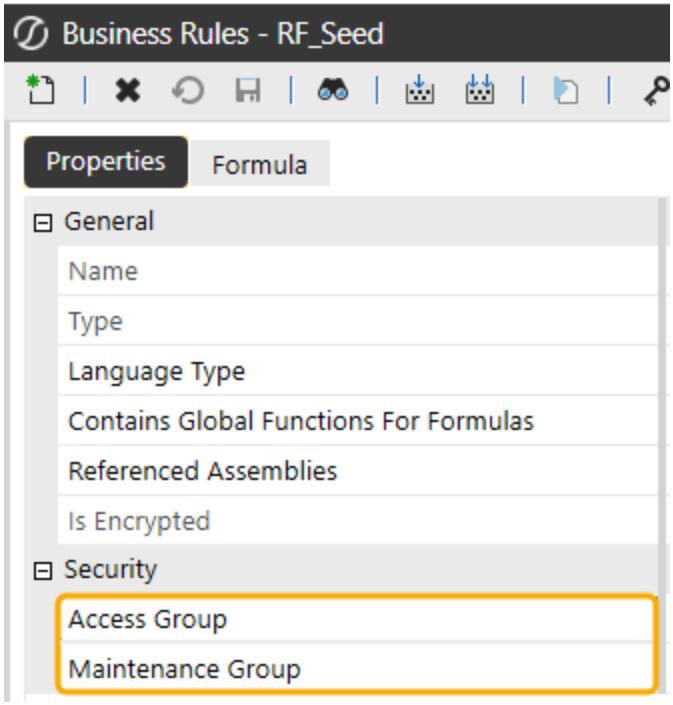

Developer Studio Setup
Completing Developer Studio Setup
Developer Studio has been installed on your computer.
Click Finish to close Setup.
Run Developer Studio
< Back
Finish
Cancel

Security Roles

These security roles are necessary to complete Developer Studio functions. Set these roles to groups that will include all Developer Studio users.

Security Role	Developer Studio Function
EncryptBusinessRules	Encrypt/Decrypt Business Rules and Workspace Assemblies
	
Workspace Assemblies	
AdministerApplicationWorkspaceAssemblies, Access Group, and Maintenance Group	Push, Pull, and Compile

Security Role	Developer Studio Function
  **NOTE:** Access Group and Maintenance Group selections can be set at both the Workspace and the Maintenance Unit level on the Workspaces page.	
Business Rules	
Access Group and Maintenance Group	Push, Pull, and Compile

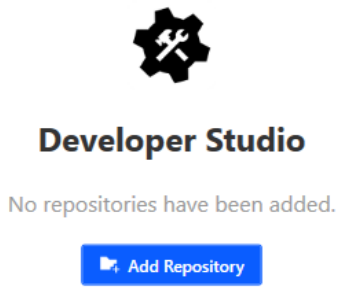
Security Role	Developer Studio Function
	

Repository Setup

A repository is a folder location on your local computer. Within that folder location, you will set up one or multiple projects to sync. In a typical setup, a repository represents the projects (business rules and workspace assemblies) that belong in a single solution. However, you can choose to use one repository for many solution projects.

To get started in Developer Studio, complete the following steps to create a repository:

1. Click the **Add Repository** button or go to **File** > **Add Repository**.

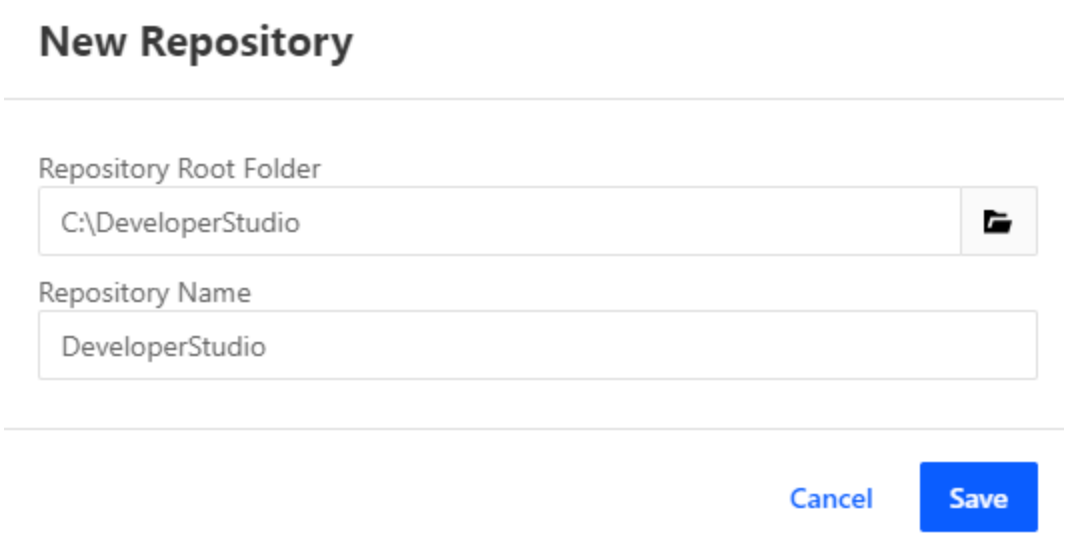


2. In the **New Repository** dialog box, click the **Browse** icon and use **File Explorer** to select a repository root folder.

> **TIP:** It is recommended that you select an empty folder the first time Developer Studio is run.

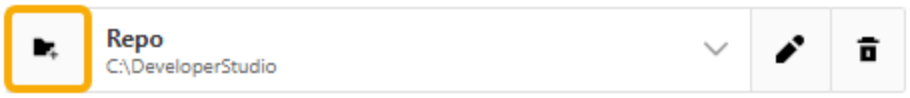
3. Enter a name for the repository.

4. Click the **Save** button.



Repositories

The home screen displays once a repository is made. On the home screen, you can manage repositories and cycle between them. You do not need to be connected to a server to manage repositories. To connect to a server and begin adding items, see Connect to a Server and Manage Items.

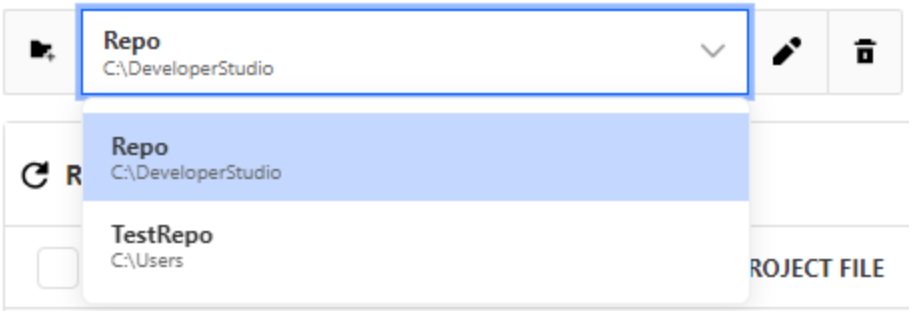


To add repositories, click the **Add Repository** icon. You can also go to **File** > **Add Repository**. Name the repository and select a root folder. The repository path you choose is where your project files will be added to. When you add a new repository, the home screen defaults to that repository.

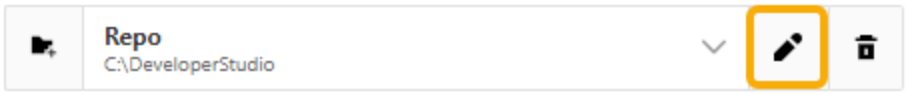
> **NOTE:** The repository name must be unique across your local machine.

To set a default repository root folder, go to **File** > **Settings**. The folder specified here will display as the default repository root folder when creating a new repository. See Settings.

All saved repositories display for selection in the drop-down menu.

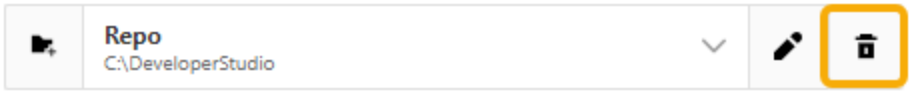


> **TIP:** You can create a parent repository to house all your repositories and display a high-level view of your repository items. For example, you can name your repository All Repositories and set it to the root folder location. When this parent repository is selected, all items within repositories under the parent will display in the grid.

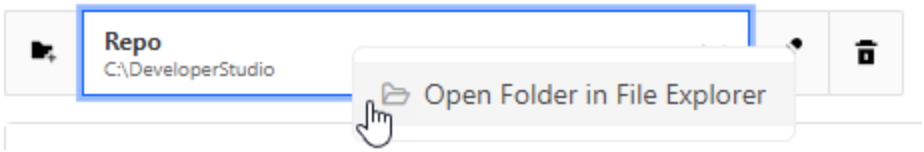


To edit the repository, select it from the drop-down menu and click the **Manage Repository** button. In the **Manage Repository** dialog box, you can change the folder location and rename the repository.

> **NOTE:** Changing the folder location does not move your items to that folder. It changes which files the repository points to.



To remove a repository, click the **Remove Repository** button. Removing a repository does not delete content from your local or within OneStream. Only the repository reference is deleted.



To view the folder location of the selected repository, right-click on the repository object and select **Open Folder in File Explorer**.

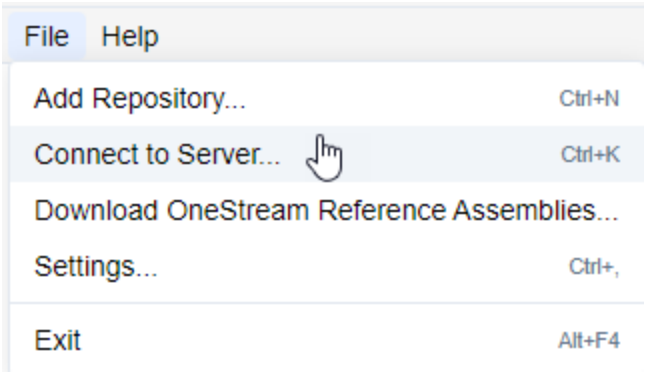
If a repository folder has been altered or deleted, the repository item will display an error in the drop-down menu to indicate that the repository location does not exist.

Connect to a Server

You need to connect to a server to create items and manage OneStream business rules and workspace assemblies. The red Connection icon on the home screen indicates that you are not connected to a server.

Complete these steps to connect to a server:

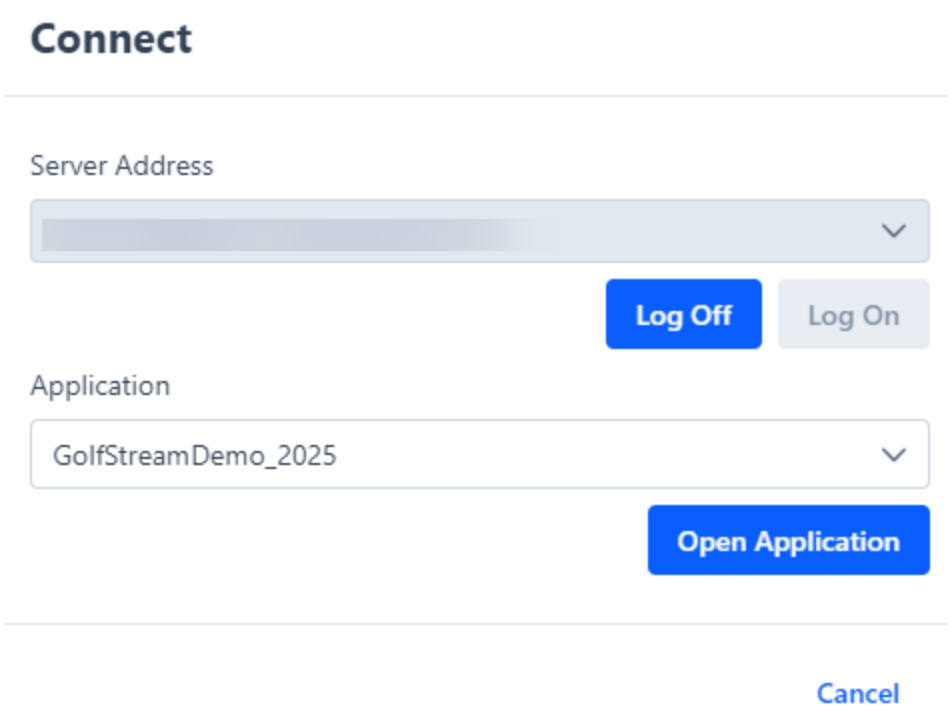
1. Go to **File** > **Connect to Server** or click **Connect**.



2. In the **Connect** dialog box, enter a server address. If you have previously connected to a server, you can select the server address from the drop-down menu.

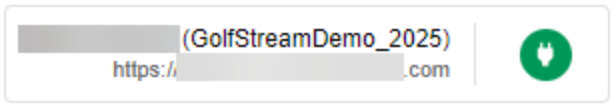
> **NOTE:** Remove OneStreamWeb from the server address. For example, if your server address is https://my.onestreamserver.com/OneStreamWeb, enter https://my.onestreamserver.com.

3. Click the **Log On** button and complete your OIS logon process.
4. From the **Application** drop-down menu, select an application and then click the **Open Application** button.

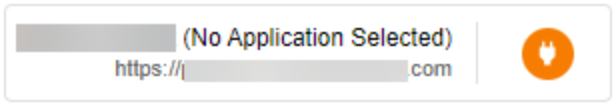


> **NOTE:** The most recently opened application on the server is saved and automatically selected in the Application field.

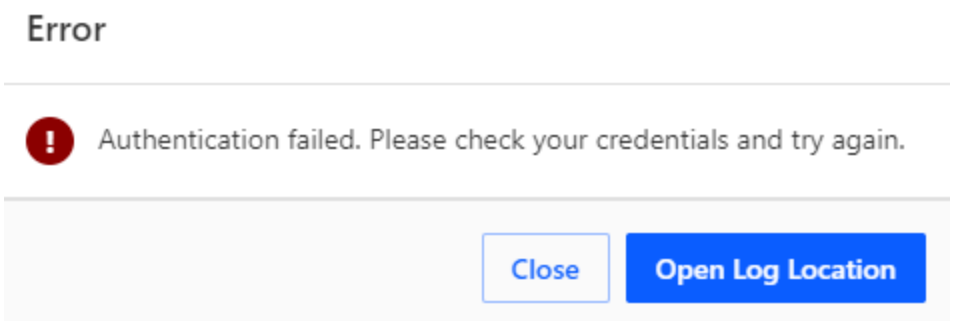
The home screen now displays a green Connection icon to indicate you are connected to a server and application. The tool bar displays the user, application, and server URL.



If you are connected to a server but have not selected an application, the Connection icon will be orange.



If an exception occurs on the server, you will receive an error. Click the **Open Log Location** button to review logging information about the exception.



NOTE: Log files are located at AppData\Local\OneStream\Developer Studio\logs.

Complete these steps to switch to a different application:

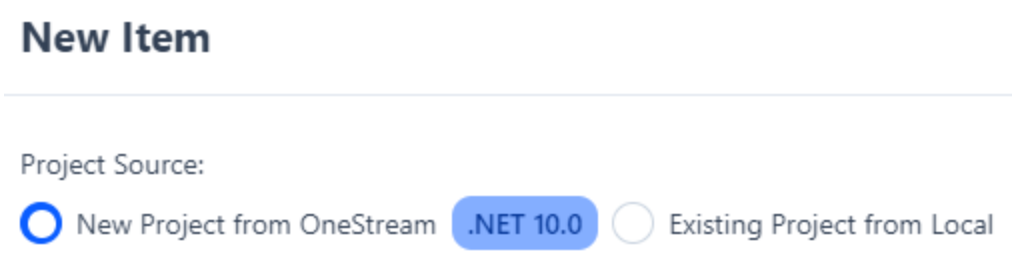
1. Go to **File** > **Connect to a Server**, or select the panel containing the server address and Connection icon to open the **Connect**dialog box.
2. From the **Application** drop-down menu, select a different application.
3. Click the **Open Application** button.

Manage Items

An item is a business rule or workspace assembly from OneStream that you can pull to your local. Once you have added an item, you can add, edit, delete, and rename files and folders from the IDE of your choice and sync those changes with the server.

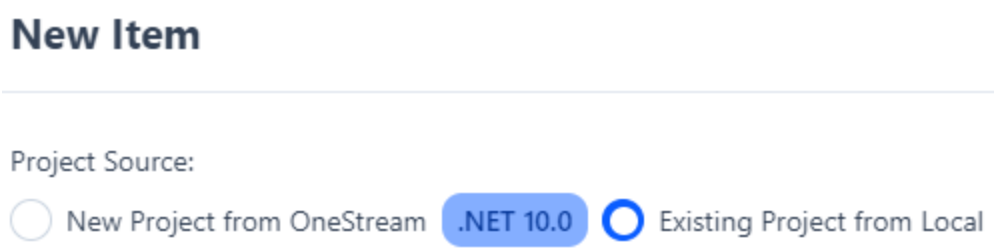
There are two ways you can create items:

- If you do not have a .NET project defined locally and want to create one based on what you have in OneStream, your Project Source selection should be New Project from OneStream. This selection creates a .NET project file, .csproj or .vbproj, and pulls down the selected OneStream files. See New Project From OneStream.



- If you have an existing project locally and want to map it to a business rule or workspace assembly from OneStream to manage, your Project Source selection should be Existing

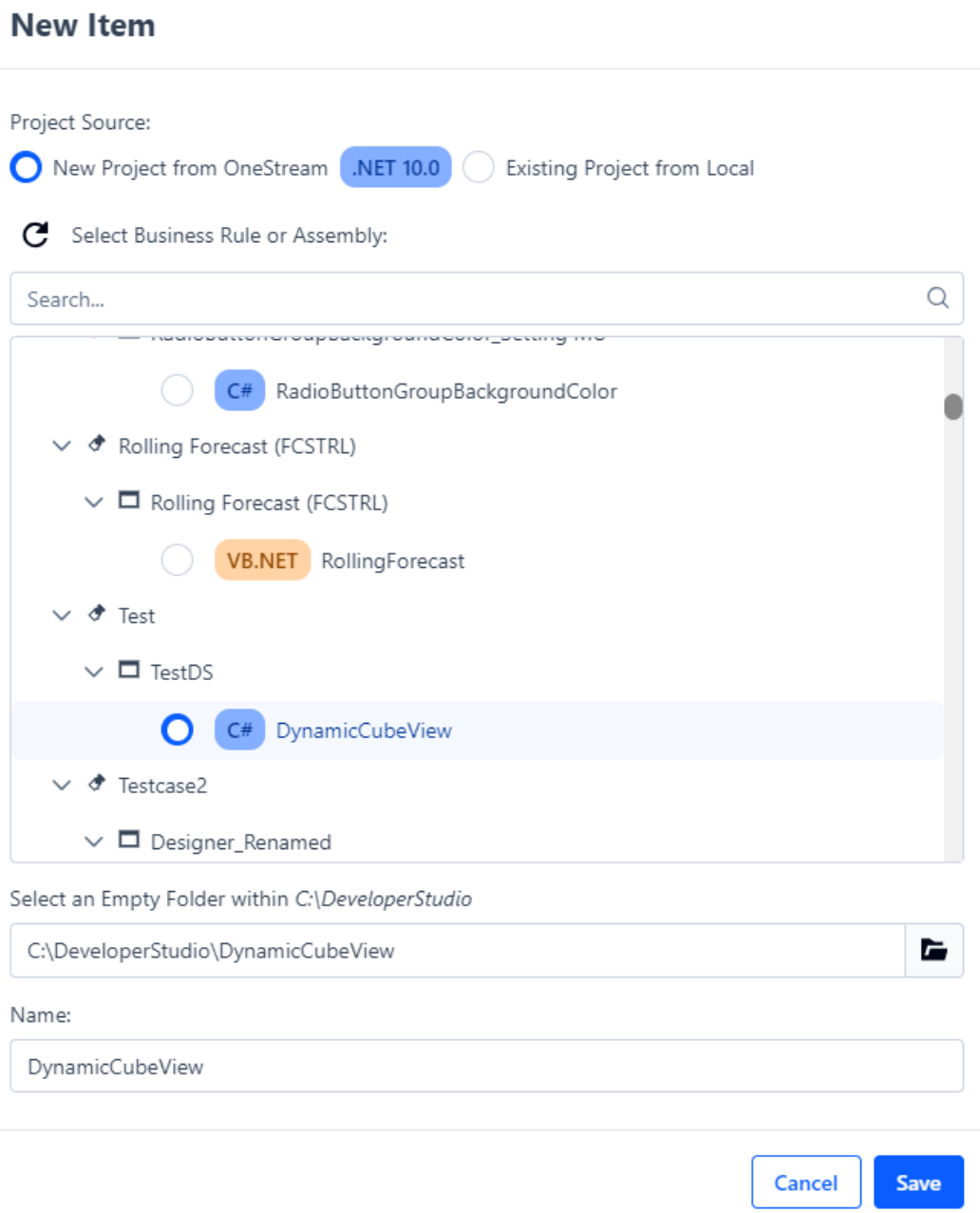
Project from Local. See Existing Project From Local.



New Project From OneStream

Complete these steps to add an item and create a new project from OneStream.

1. Click the **Add Item** button to display the **New Item** window.
2. Under **Project Source**, select **New Project from OneStream**.
3. In the navigation tree, select a business rule or workspace assembly. The project folder you will select maps to the business rule or workspace assembly selected. The navigation tree displays the compiler language for each item.

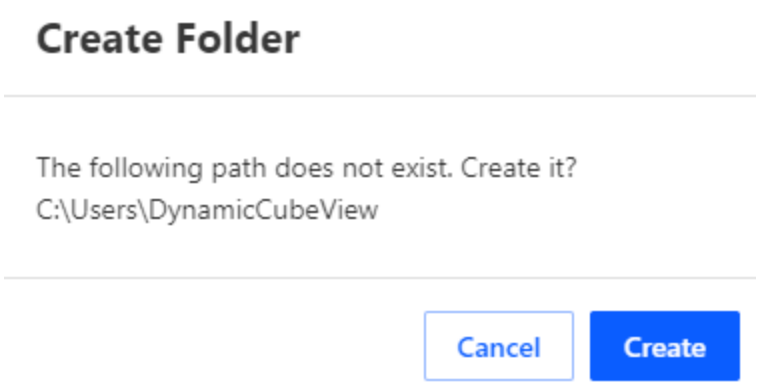

New Item
Project Source:
New Project from OneStream
.NET 10.0
Existing Project from Local
Select Business Rule or Assembly:
Search...
C#
RadioButtonGroupBackgroundColor
Rolling Forecast (FCSTRL)
Rolling Forecast (FCSTRL)
VB.NET
RollingForecast
Test
TestDS
C#
DynamicCubeView
Testcase2
Designer_Renamed
Select an Empty Folder within C:\DeveloperStudio
C:\DeveloperStudio\DynamicCubeView
Name:
DynamicCubeView
Cancel
Save

To refresh business rules and workspace assemblies within the navigation tree, click the **Refresh** button.

Items with this icon are encrypted and cannot be selected until they are decrypted. See New Project From OneStream.

4. The Select an Empty Folder field defaults to the repository root path. After selecting a business rule or workspace assembly, the field updates automatically to include a subfolder with the item name. To change the folder, click the **Browse** icon and use **File Explorer** to select an empty folder in that repository.

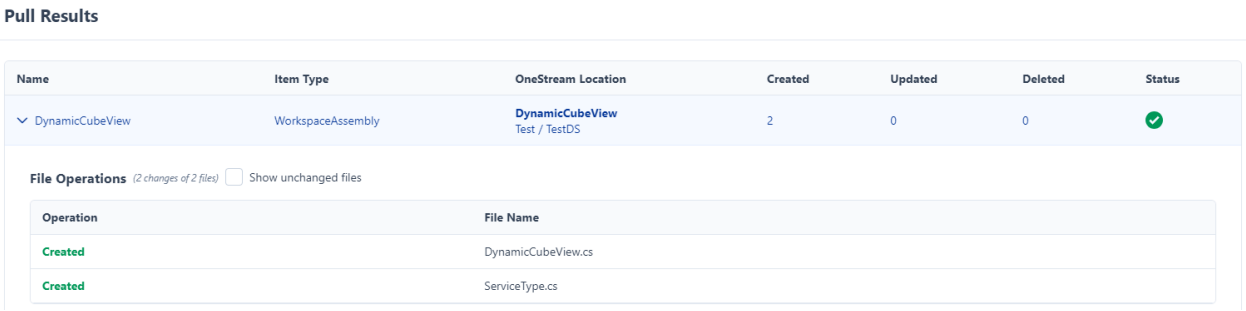
   If the selected folder does not exist, the Create Folder dialog box displays when you click the Save button. Click the **Create** button to create a folder with the name of the assembly or business rule selected.



> **NOTE:** You will receive an error if the folder you selected is not empty or not within the repository directory.

5. The Name field defaults to the selected assembly or business rule name. Rename the item if needed. The item name is local to Developer Studio and does not affect the OneStream server in any way.
6. Click the **Save** button.

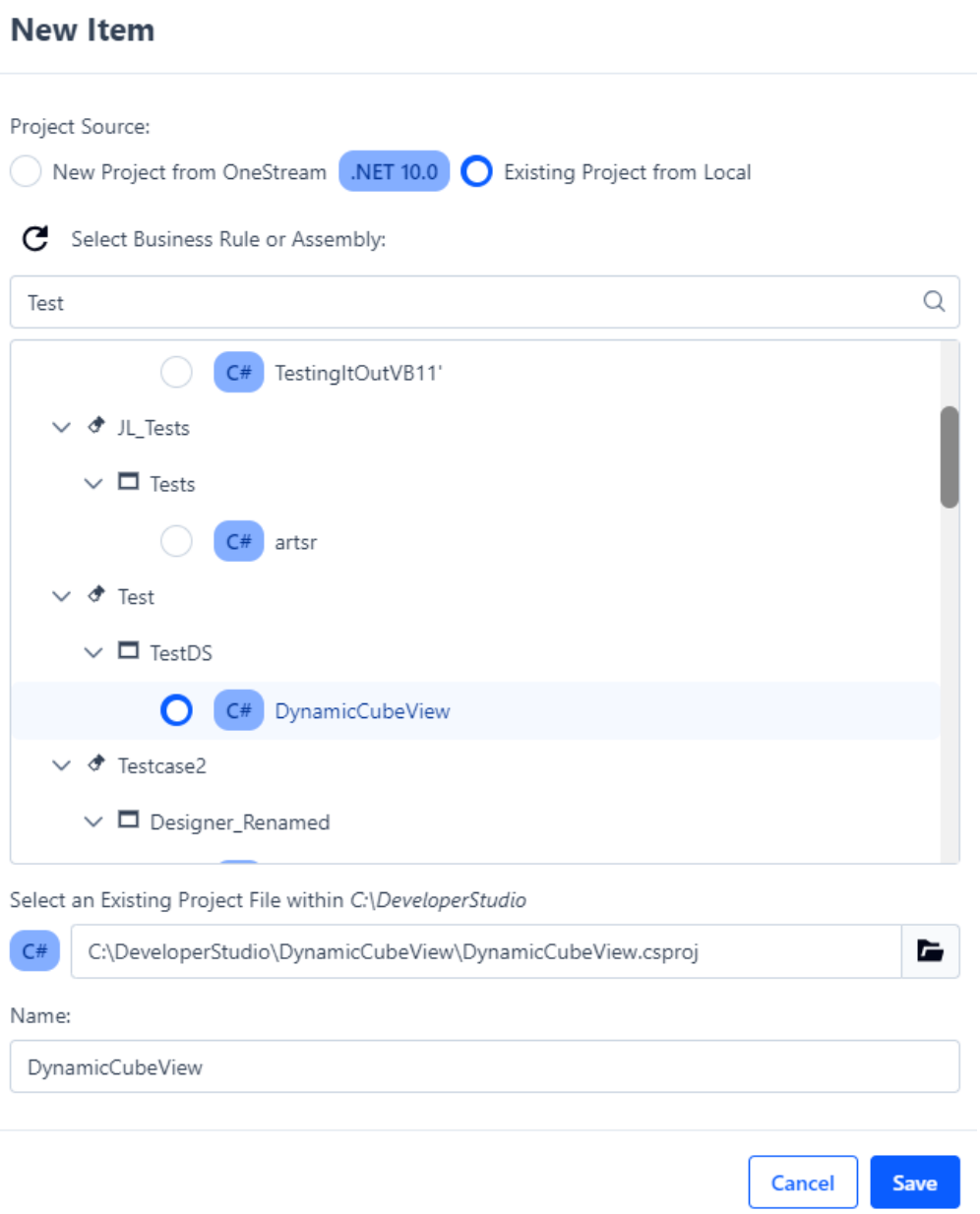
The Pull Results window will display the item and its files that were pulled down from OneStream to your local. Click the item to view the operation and file name for each file change. See Push and Pull Results.



Existing Project From Local

Complete these steps to add an item to an existing project from local:

1. Click the **Add Item** button to display the **New Item** window.
2. Under **Project Source**, select **Existing Project from Local**.
3. In the navigation tree, select a business rule or workspace assembly. The navigation tree only displays items with the same compiler language as the project file you select. All local files within the project file you select will map to the business rule or workspace assembly selected from the navigation tree.



To refresh business rules and workspace assemblies, click the **Refresh** icon.

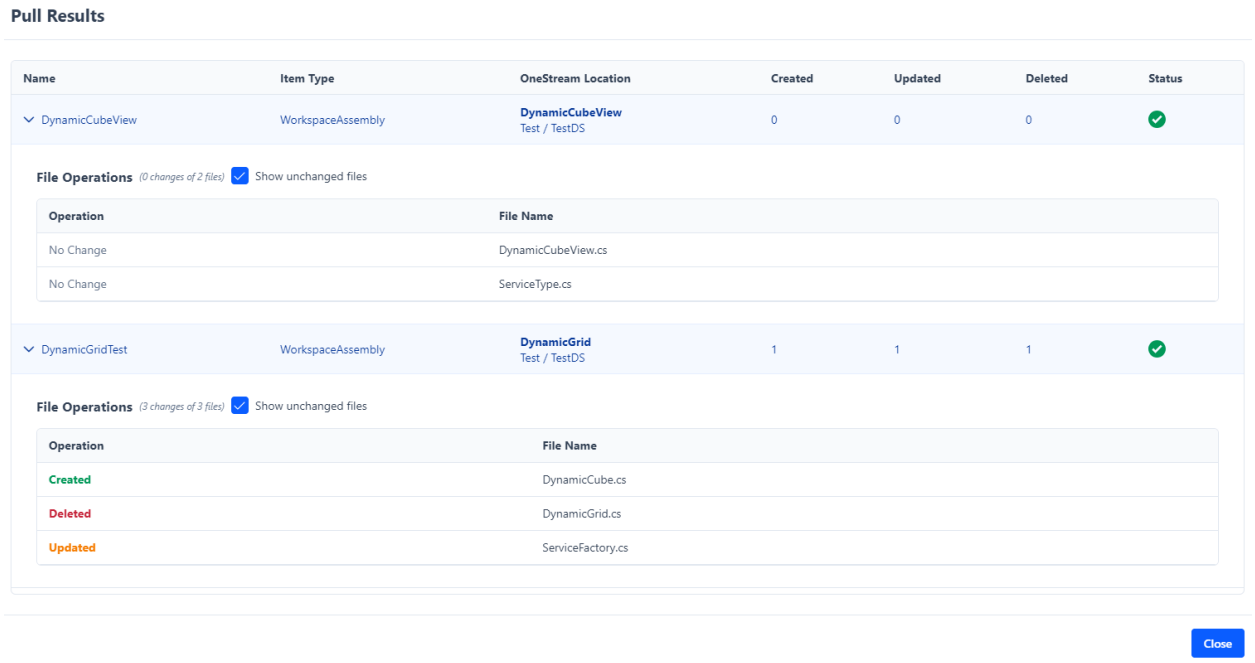
 Items with this icon are encrypted and cannot be selected until they are decrypted. See Existing Project From Local.

4. Click the **Browse**  icon and use **File Explorer** to select an existing project file within your repository. The compiler language displays next to the file path after selecting a project file.

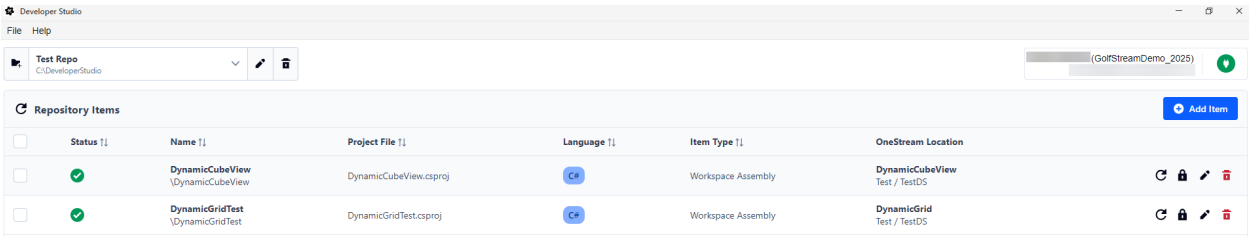
   > **NOTE:** The project file must exist within the repository directory.

5. The Name field defaults to the project file name. Rename the item if needed. The item name is local to Developer Studio and does not affect the OneStream server in any way.
6. Click the **Save** button.
7. The Pull Latest Code dialog box displays when you save the item.
   a. To overwrite your local project files with the chosen business rule or workspace assembly files, click the **Pull Latest** button.
   b. To keep what you have locally in your project file, click the **Do Not Pull** button.

If you selected Pull Latest, the Pull Results window displays showing the item that was pulled to your local and all files created, updated, or deleted. Click the item to view the operation and file name for each file change. See Push and Pull Results.



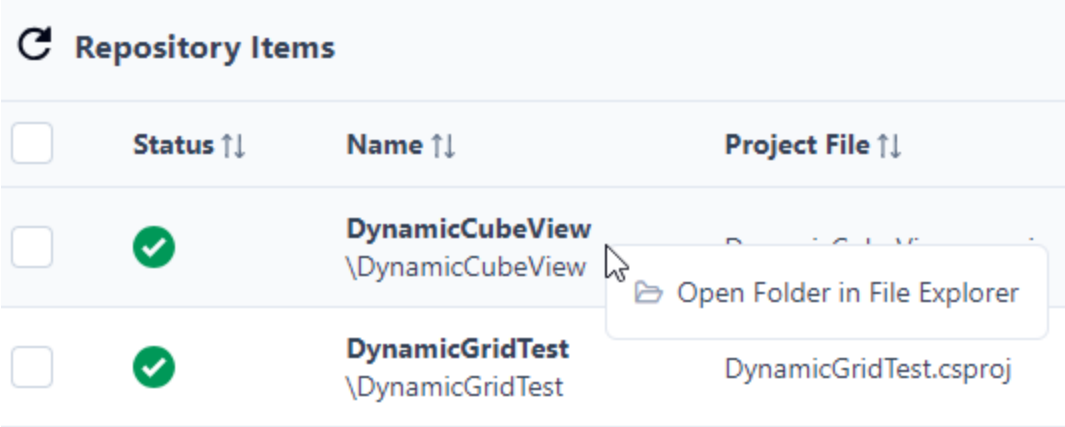
Repository Items Grid



All items within the currently selected repository display in the Repository Items grid with the following columns:

Column	Description
Status	The status of the item:  **Ready**: The item is valid and ready to push and pull.  **Conflicts**: The item has conflicts with the server that must be resolved before you can push and pull. For example, if the item you are trying to push has been encrypted on the server. Select the status icon to view the specific conflicts.  **Error**: The item has errors that must be resolved before you can push and pull. For example, if the business rule or workspace assembly does not exist on the server. Select the status icon to view the issues.
Name	The name of the item and the full file path.
Project File	The name of the project file.
Language	The languages of the project file, which can be C# or VB.NET.
Item Type	The item type is either business rule or workspace assembly.
OneStream Location	The location of the item within OneStream. For example, WorkspaceA - MUA - Assembly A. WorkspaceA is the Workspace name, MUA is the Maintenance Unit name, and Assembly A is the assembly name.

To view the folder file location of an item, right-click on the item and select **Open Folder in File Explorer**.



To manually refresh the Repository Items grid, click the **Refresh** icon.

> **NOTE:** If you edit your osmap files outside of Developer Studio, the grid will automatically refresh.

To manually refresh a single item, click the **Refresh** icon in the item's row.

To encrypt or decrypt an item, click the **Encrypt/Decrypt** icon. See Repository Items Grid.

To edit an item, click the **Edit** icon. In the Edit Item window, you can select a new business rule or workspace assembly from OneStream. You cannot rename an item or edit the selected project file.

To remove an item, click the **Remove** icon. You cannot multiselect items for deletion.

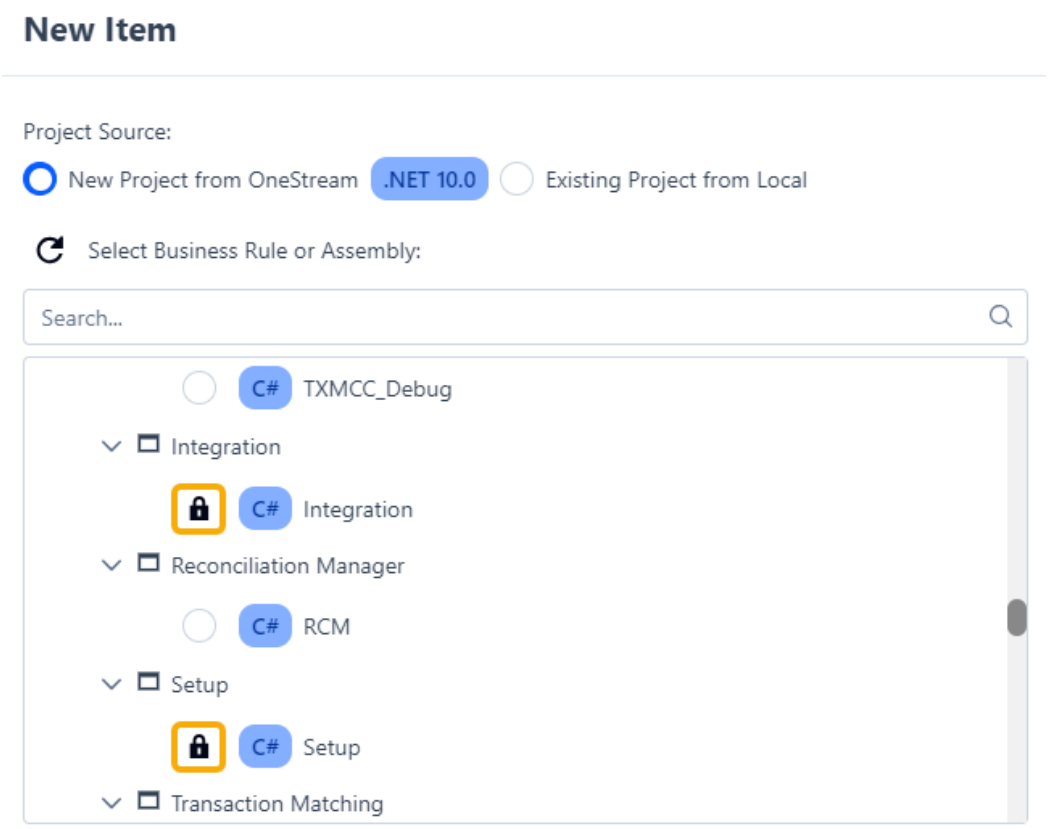
> **IMPORTANT:** When you remove an item, nothing is deleted locally or on the server. If you want to delete your project or source file, this must be done manually outside of Developer Studio.

> **NOTE:** If you have the same item created in two different repositories with different folder paths, removing the item from one repository will not remove it from the other.

Encrypt and Decrypt Items

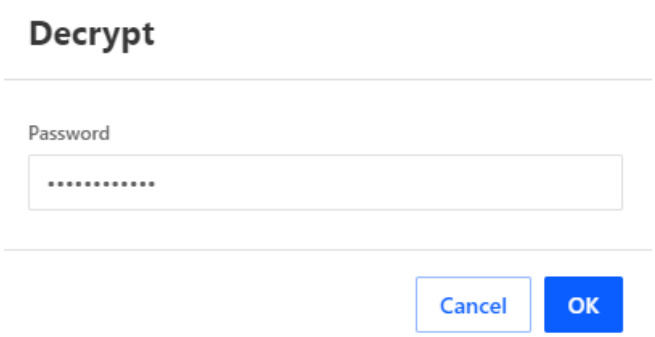
Before you can work with an item in Developer Studio, all its contents must be available unencrypted on the server. To decrypt a file before pulling it to your local, complete these steps:

1. In the **New Item** or **Edit Item** window, select the **Decrypt** icon for an encrypted item.

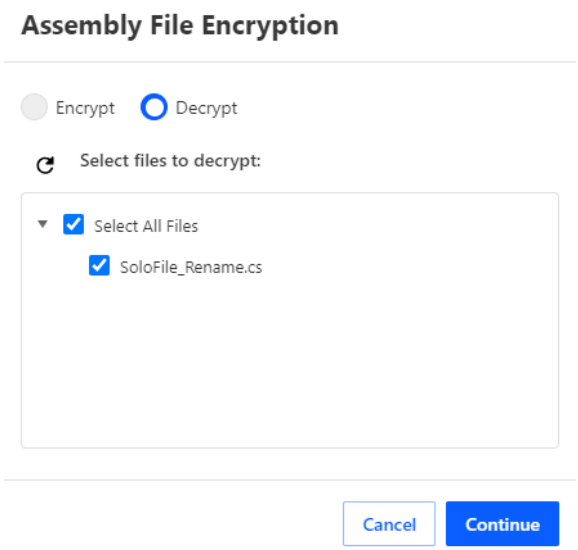


2. The window that displays depends on whether you are decrypting a business rule or a workspace assembly file.

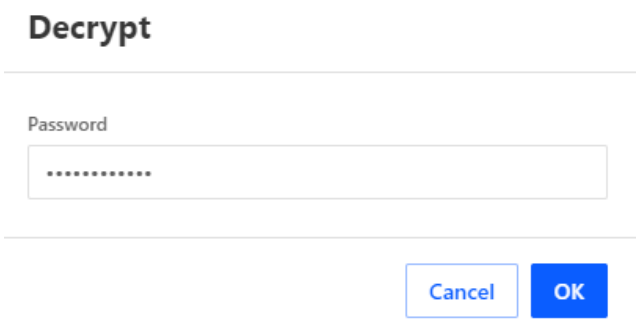
a. If you are decrypting a business rule, the Decrypt dialog box displays. Enter a valid password and click the **OK** button.



b. If you are decrypting a workspace assembly file, the Assembly File Encryption dialog box displays. Use the navigation tree to select files to decrypt and click the **Continue** button.

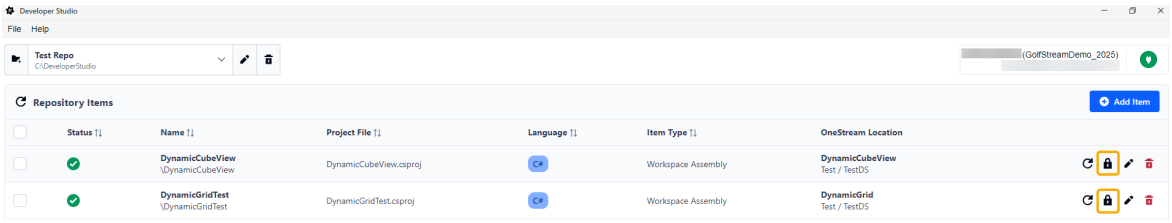


In the **Decrypt** dialog box, enter a valid password and click the **OK** button.



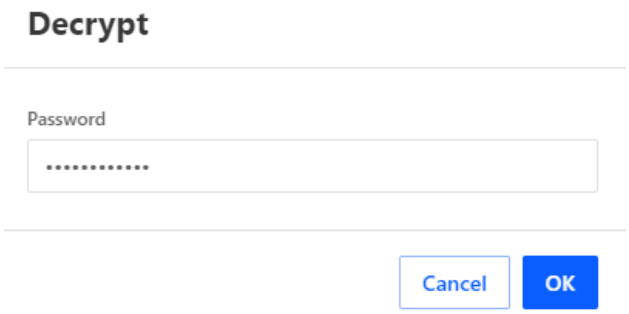
To encrypt or decrypt an item in the Repository Grid, complete these steps:

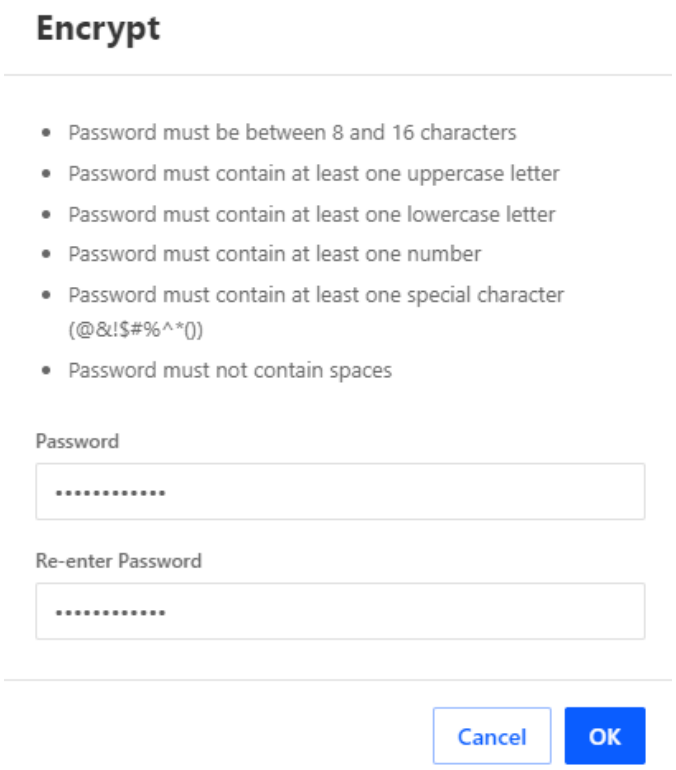
1. Click the **Encrypt/Decrypt** button for that item.



2. The window that displays depends on whether you are encrypting/decrypting a business rule or workspace assembly files.

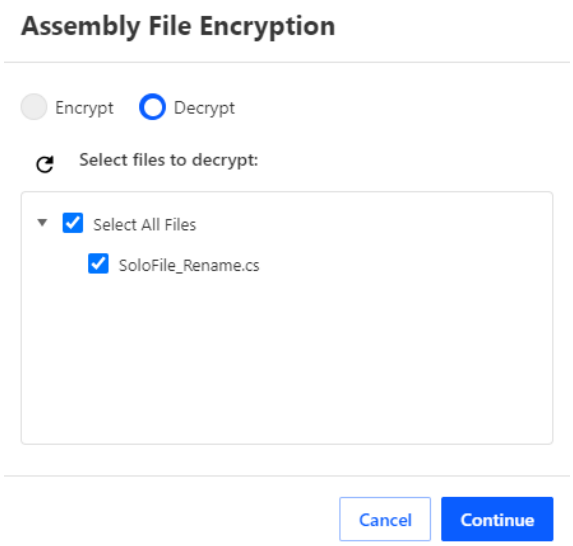
    a. If you are encrypting or decrypting a business rule, the Decrypt or Encrypt dialog box displays. Enter or create a valid password and click the **OK** button.



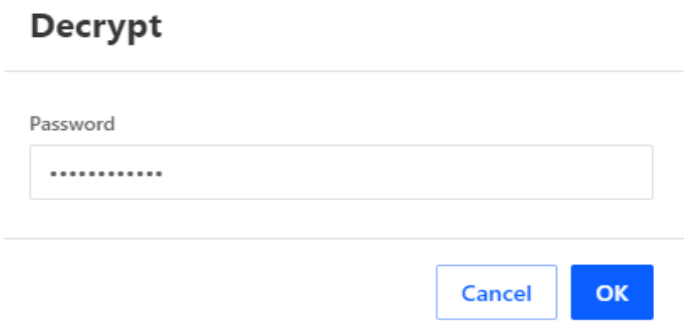


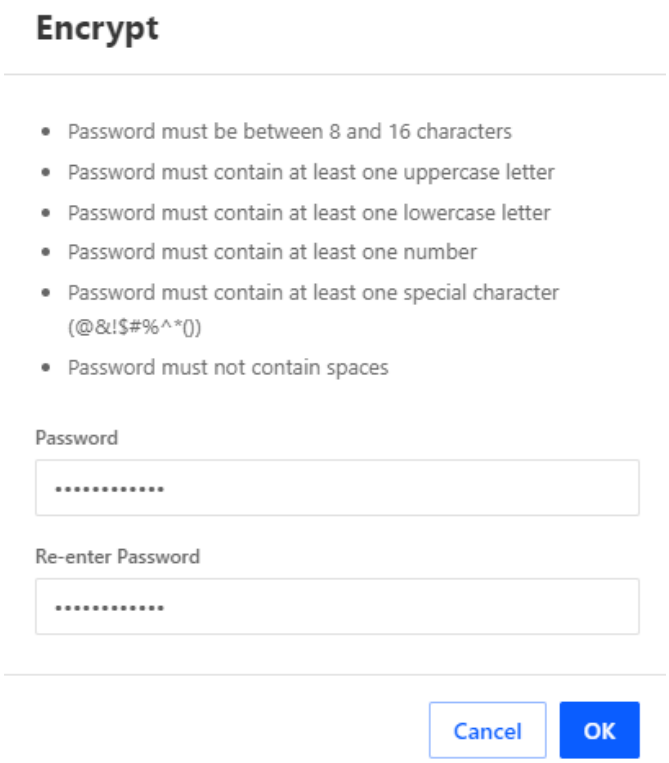
b. If you are encrypting or decrypting a workspace assembly file, the Assembly File Encryption dialog box displays. Select either **Encrypt** or **Decrypt** and then use the navigation tree to select files. Click the **Continue** button.

> **NOTE:** Once a single file in an assembly is encrypted, you cannot synchronize any file in the assembly with Developer Studio.



Enter or create a valid password and click the **OK** button.




Encrypt
Password must be between 8 and 16 characters
Password must contain at least one uppercase letter
Password must contain at least one lowercase letter
Password must contain at least one number
Password must contain at least one special character (@&!$#%^*())
Password must not contain spaces
Password
Re-enter Password
Cancel
OK

Push and Pull

Pushing and pulling enables you to keep your local files and your OneStream files in sync. Performing a push replaces the items on the OneStream server with your local content. Performing a pull replaces items on your local machine with OneStream server content.

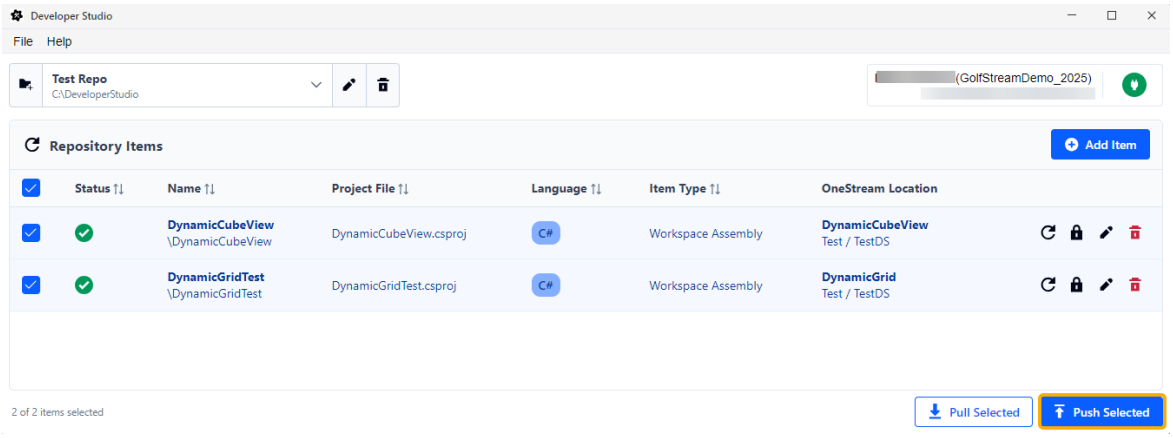
Pushing and pulling only impacts compilable files, such as C# or VB.NET files. If you have a non-compilable file in your local project, pulling from the server will not impact that file.

> **NOTE:** Items with the status Conflicts or Errors cannot be pushed or pulled. Select the status icon in the grid to view issues with the item.

Push

To push your changes to OneStream, complete these steps:

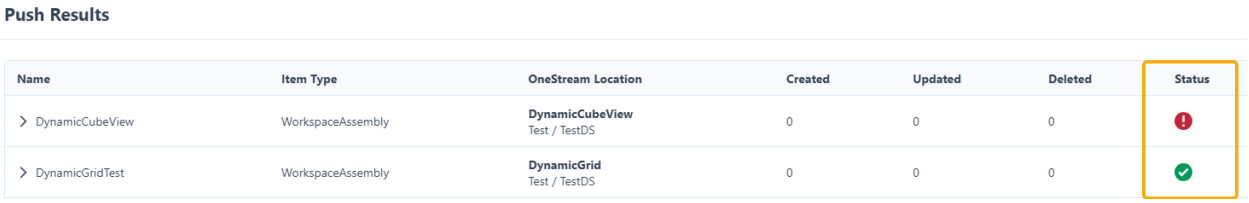
1. Complete and save the changes in your local files.
2. In the **Repository Items** grid, use the checkboxes to select the items you want to push. To select all items in the repository, select the multiselect checkbox.
3. Click the **Push Selected** button.



4. The **Confirm Push Operation** dialog box displays. To confirm that you want to replace items on the OneStream server with your local content, click the **OK** button.

   To prevent the Confirm Push Operation dialog box from displaying again, select the **Don't show this confirmation again for push and pull operations**checkbox.

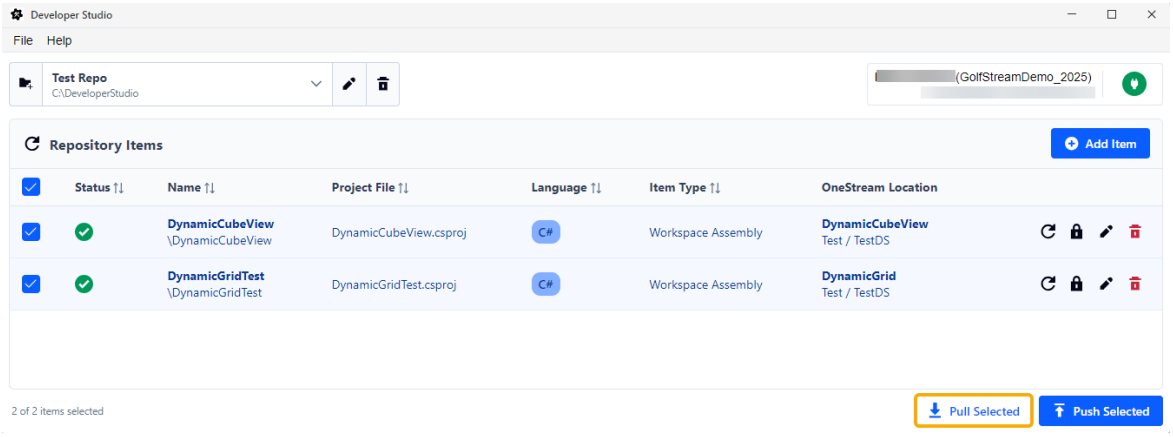
Once a push is completed, the Push Results window displays showing updates to the files. A push compiles on the server, and the Status column in the grid indicates the compilation results. See Push and Pull Results.



Pull

To pull content from OneStream down to your local, complete these steps:

1. Complete and save your changes in OneStream.
2. Use the checkboxes to select the items you want to pull these changes into. To select all items in the repository, select the multiselect checkbox.
3. Click the **Pull Selected** button.

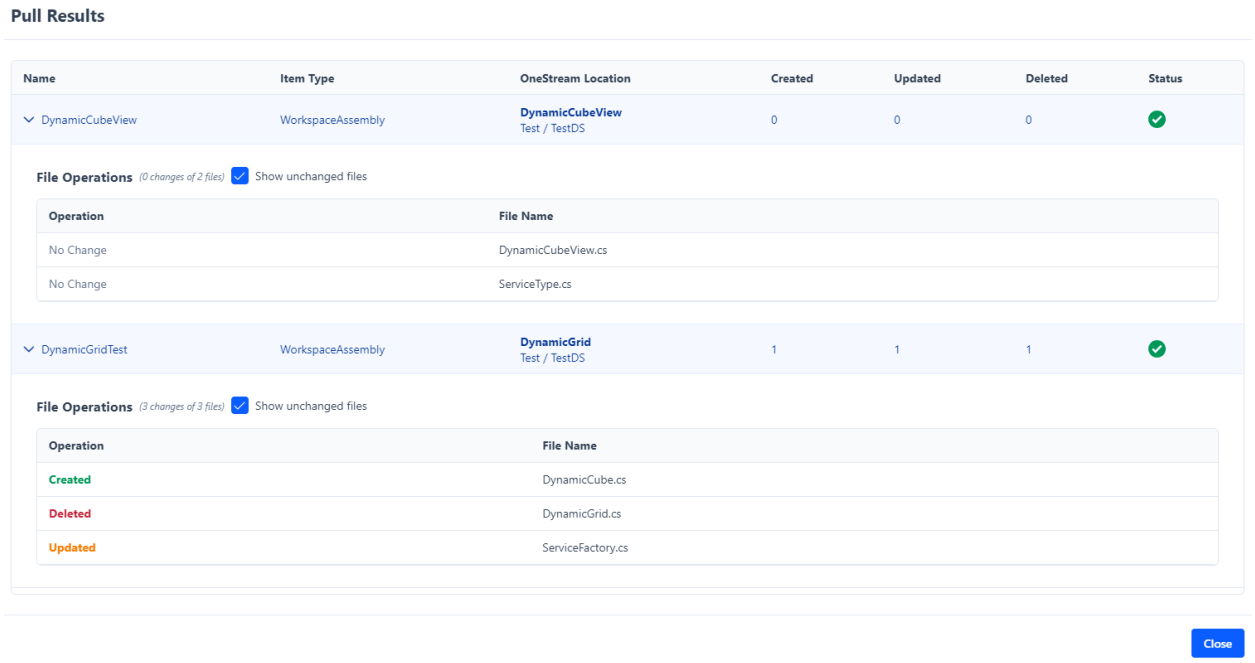


4. The **Confirm Pull Operation** dialog box displays. To confirm that you want to replace items on your local machine with OneStream server content, click the **OK** button.

   To prevent the Confirm Pull Operation dialog box from displaying again, select the **Don't show this confirmation again for push and pull operations** checkbox.

   > **CAUTION:** If you have added a file locally to any of the selected project items that have not been pushed to the server, performing a pull will delete this local file.

Once a pull is completed, the Pull Results window displays showing updates to your local files. Click the item to view the operation and file name for each file change. See Push and Pull Results.



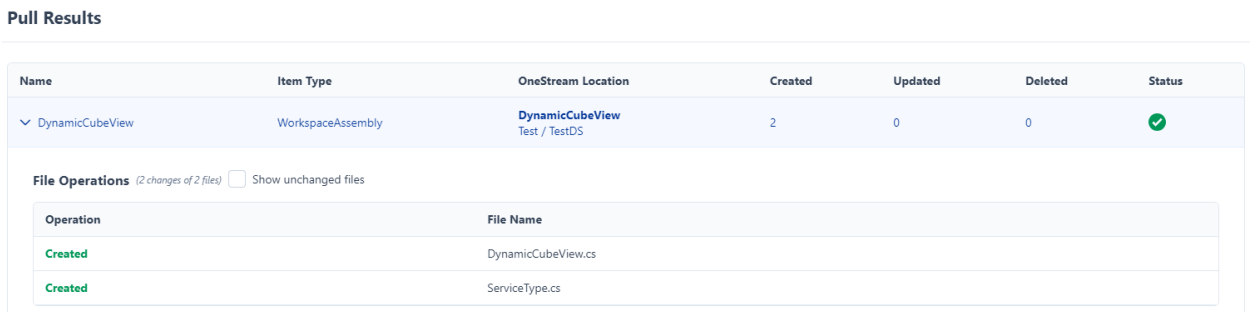
Push and Pull Results

The Push/Pull Results window displays whenever items are created, pushed, or pulled. It contains the following information:

- Each item that was affected
- The status of each item
- All files that were created, deleted, or updated
- Validation errors
- Compile Results that list all errors and warnings for push operations

> **NOTE:** Compile Results do not display for pull operations.

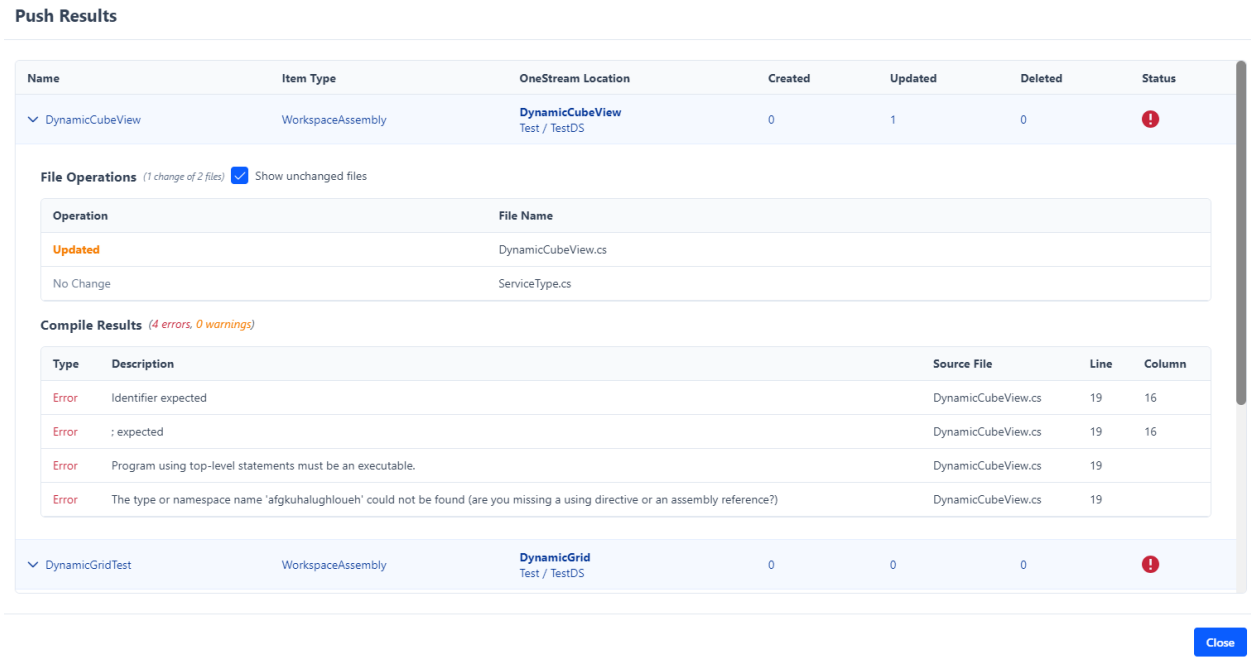
The grid contains a high-level overview of each item. It lists the item name, type, OneStream location, status, and the number of files created, updated, and deleted.



The Status column displays an icon for the item's status:

- **Success**: The item was pushed or pulled successfully.
- **Warnings**: There are warnings for the item. View all warnings in Compile Results.
- **Errors**: There are errors for the item. View all errors in Compile Results.

Select the item to drill down into that item and view additional information.

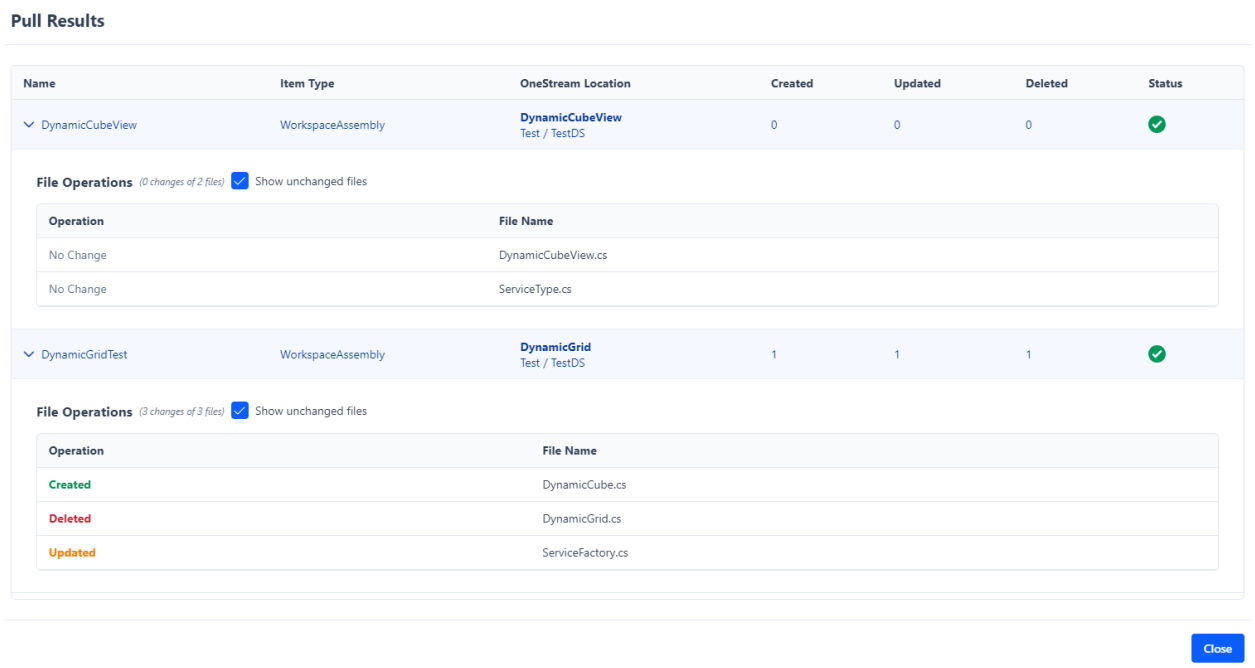


File Operations

By default, only the files that were changed in the operation display in the grid. To display all files in the operation, select the **Show unchanged files** checkbox.

The Operation column entries are color coded:

- **Created**: This entry is colored green and indicates a file was created.
- **Deleted**: This entry is colored red and indicates a file was deleted.
- **Updated**: This entry is colored orange and indicates a file was updated.
- **No Change**: This entry is colored gray and indicates that no changes were made to the file.



Compile Results

This grid details any warning or errors associated with the push operation.

The Type column entries are color coded:

- **Warning**: This entry is colored orange and indicates the item has a compilation warning.
- **Error**: This entry is colored red and indicates the item has a compilation error.

Push Results

Name	Item Type	OneStream Location	Created	Updated	Deleted	Status
DynamicCubeView	WorkspaceAssembly	**DynamicCubeView** Test / TestDS	0	1	0	

File Operations *(1 change of 2 files)* Show unchanged files

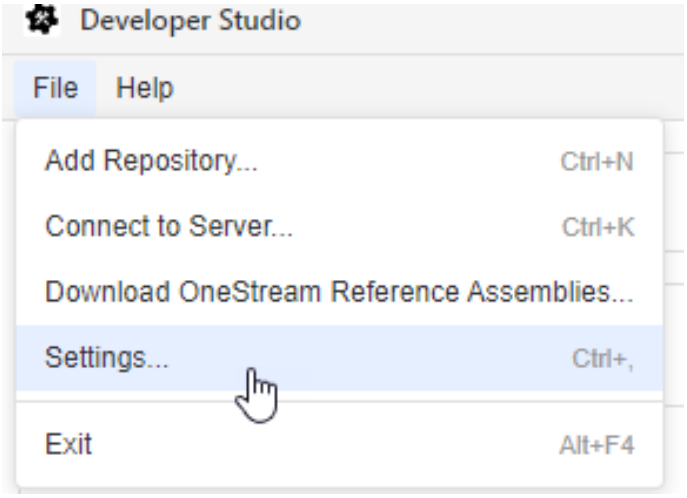
Operation	File Name
Updated	DynamicCubeView.cs

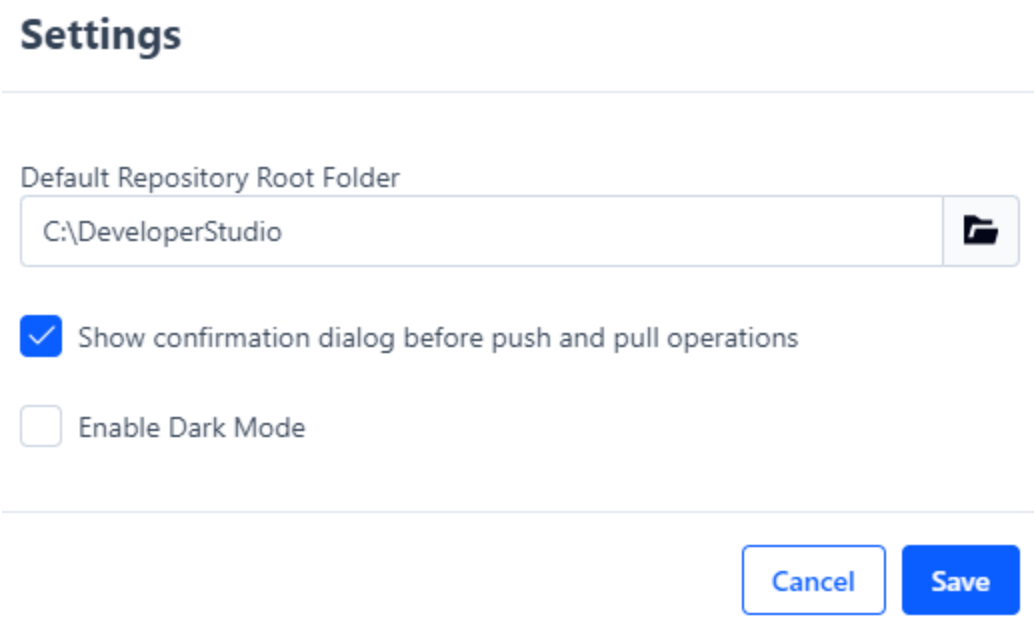
Compile Results *(4 errors, 1 warnings)*

Type	Description	Source File	Line	Column
Error	Syntax error, ',' expected	DynamicCubeView.cs	19	10
Error	; expected	DynamicCubeView.cs	19	20
Error	Program using top-level statements must be an executable.	DynamicCubeView.cs	19	
Error	The type or namespace name 'Test' could not be found (are you missing a using directive or an assembly reference?)	DynamicCubeView.cs	19	
Warning	The variable 'test' is declared but never used	DynamicCubeView.cs	19	5

Settings

Navigate to **File** > **Settings**.



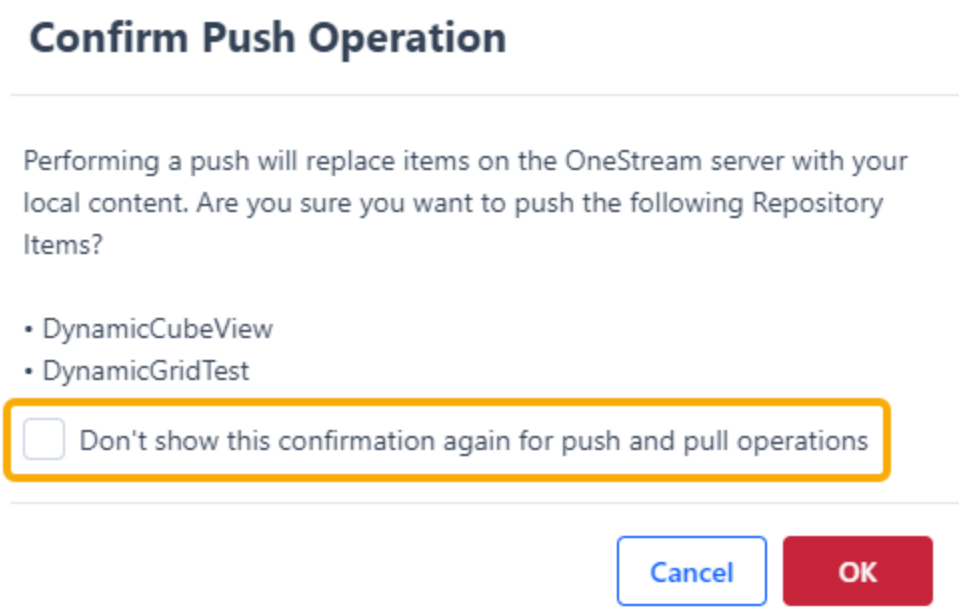


Default Repository Root Folder

To set a default repository root folder, click the **Browse** icon and use **File Explorer** to select a folder. This will display as the default repository root folder setting when you create a new repository.

Show confirmation dialog before push and pull operations

Select this checkbox to display a confirmation dialog before push and pull operations. If you previously selected the Don't show this confirmation again for push and pull operations checkbox, selecting this checkbox in Settings restores the confirmation dialog box for those operations.



Enable Dark Mode

Select this checkbox to enable dark mode.

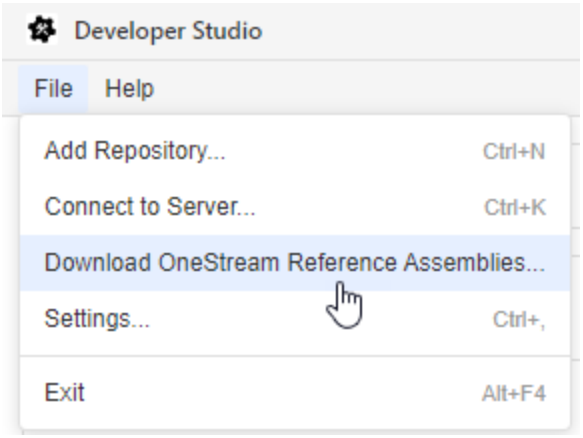
OneStream Reference Assemblies

You can download OneStream reference assemblies to integrate IntelliSense capabilities in your IDE. For information on creating a local NuGet repository, see Setting Up Local NuGet Feeds.

> **NOTE:** You must be connected to a server to download OneStream reference assemblies. See Connect to a Server.

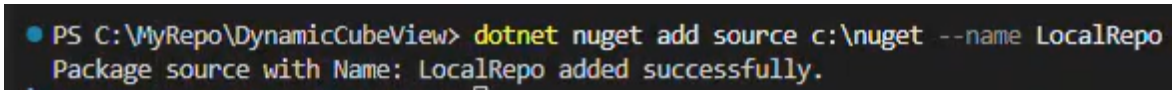
Complete these steps to download reference assemblies and configure IntelliSense:

1. Navigate to **File** > **Download OneStream Reference Assemblies**.



2. From the **Select Version** drop-down menu, select a reference assembly version.
3. Click the **Browse** icon and use **File Explorer** to select the download location and then click the **Download** button. The location defaults to your Downloads folder.
4. When the progress bar is completed, click the **Open Folder in File Explorer** button to display the NUPKG file's folder location.

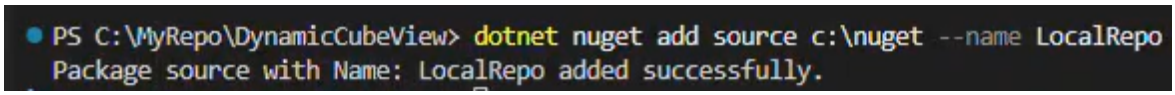
5. Open a terminal window or command prompt and then navigate to the directory containing the item or project you want to add the reference to. In the following example, the reference is added to the DynamicCubeView project.



   Enter the following command to set up a local NuGet repository for the reference assemblies:

```
dotnet nuget add source
```

   In this example, the repository points to the c:\nuget directory and is named LocalRepo.



6. Enter this command to add the reference assemblies to your project:

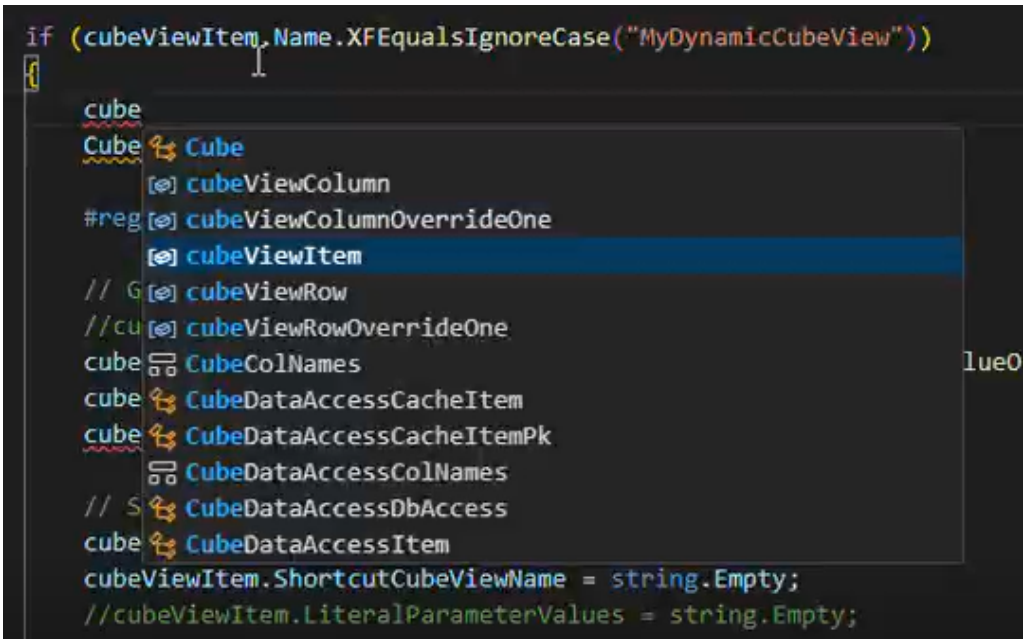
```
dotnet add package OneStream.Platform.ReferenceAssemblies
```

   This command must also include the version number listed in the package file name. In this example, the file name lists version 9.2.0, so that version is listed in the command.

> **IMPORTANT:** This second command must be entered in each project file.

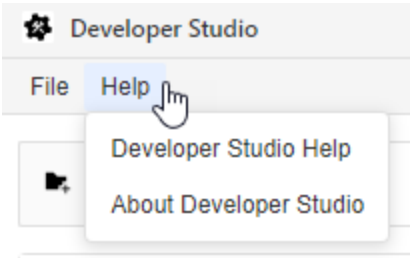
For more information on adding dotnet packages and local feeds, see Dotnet Package Add and Setting Up Local NuGet Feeds.

IntelliSense functions in your IDE once these commands are entered. This image shows IntelliSense functioning locally in Visual Studio.



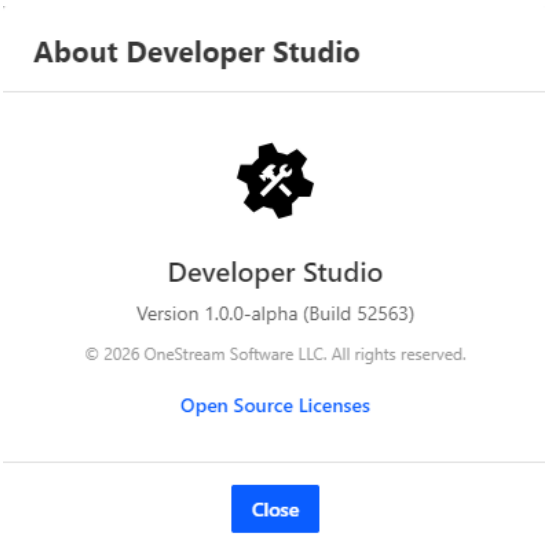
Help Options

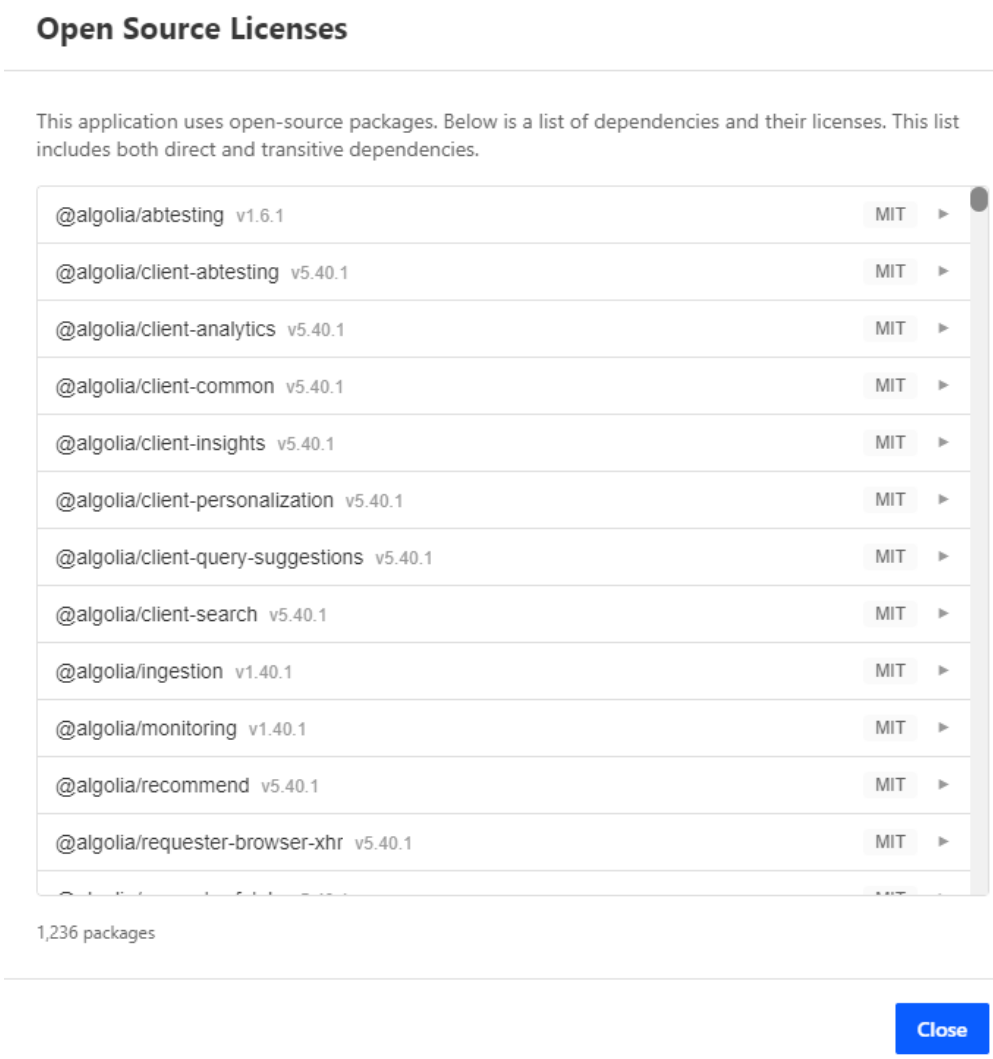
Use the Help tab to access support resources.



Select **Help** > **Developer Studio Help** to access online help documentation.

Select **Help** > **About Developer Studio** to access version information and open source licenses.




Open Source Licenses
This application uses open-source packages. Below is a list of dependencies and their licenses. This list includes both direct and transitive dependencies.
@algolia/abtesting v1.6.1 MIT
@algolia/client-abtesting v5.40.1 MIT
@algolia/client-analytics v5.40.1 MIT
@algolia/client-common v5.40.1 MIT
@algolia/client-insights v5.40.1 MIT
@algolia/client-personalization v5.40.1 MIT
@algolia/client-query-suggestions v5.40.1 MIT
@algolia/client-search v5.40.1 MIT
@algolia/ingestion v1.40.1 MIT
@algolia/monitoring v1.40.1 MIT
@algolia/recommend v5.40.1 MIT
@algolia/requester-browser-xhr v5.40.1 MIT
1,236 packages
Close

Appendix A: Third-Party Assemblies

This version of Developer Studio does not support third-party assemblies in the Reference Assemblies NuGet package. If your solution depends on these third-party assemblies for compilation, obtain them through a public NuGet feed:

Assembly	Version Compatible with Developer Studio
AutoMapper	13.0.1
DocumentFormat.OpenXml	3.2.0
HtmlTextWriter	3.0.1
Newtonsoft.Json	13.0.3
Dapper	2.1.35