

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Making Interactive Experiences

/ Data Driven Gameplay Elements

/ Data Registries

/ Data Registries Quick Start

Data Registries Quick Start

Get started with the Data Registries plugin.

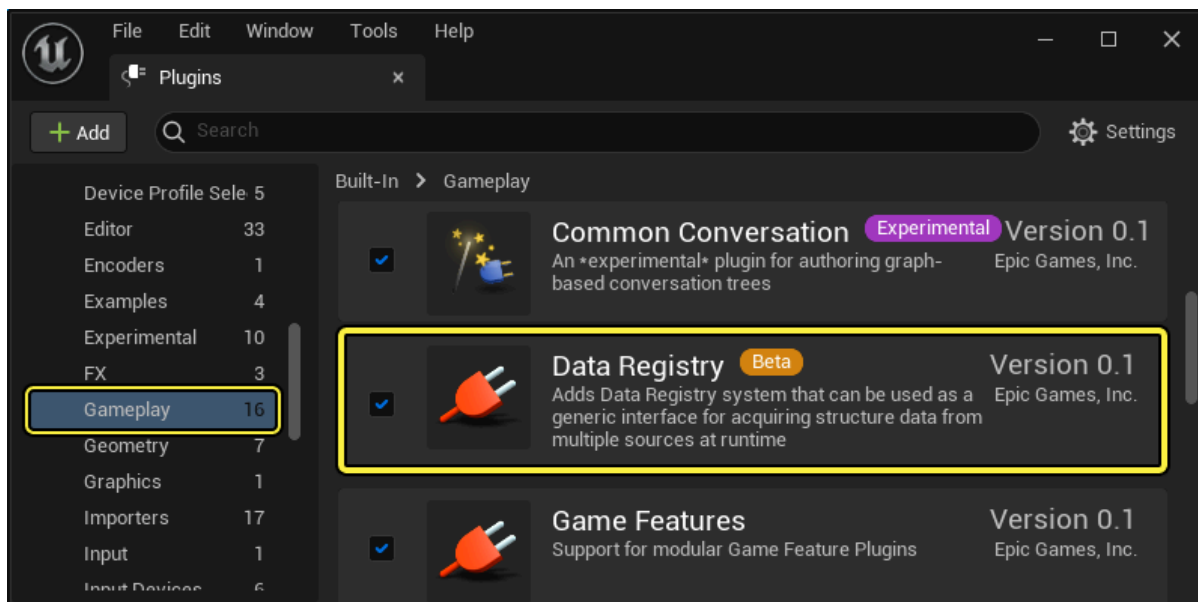


To begin using **Data Registries**, you will need to enable the Data Registry plugin, then you can create, configure, and populate your first Data Registry.

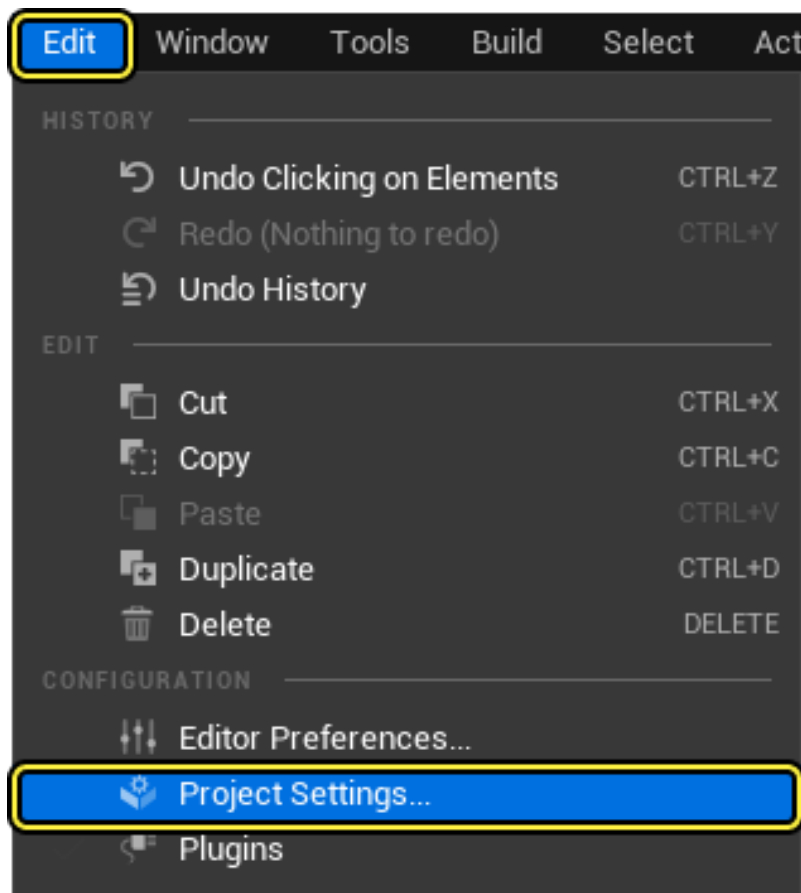
1. Enable the Data Registry plugin by navigating to **Edit > Plugins**. When the **Plugins** window appears, search or browse through the plugins to find the **Data Registry** plugin located in the **Gameplay** category. Once you have found the plugin, ensure that the **Enabled** checkbox is checked.



If the Plugins window indicates that you need to restart the editor for the plugin to take effect, do so before proceeding further.



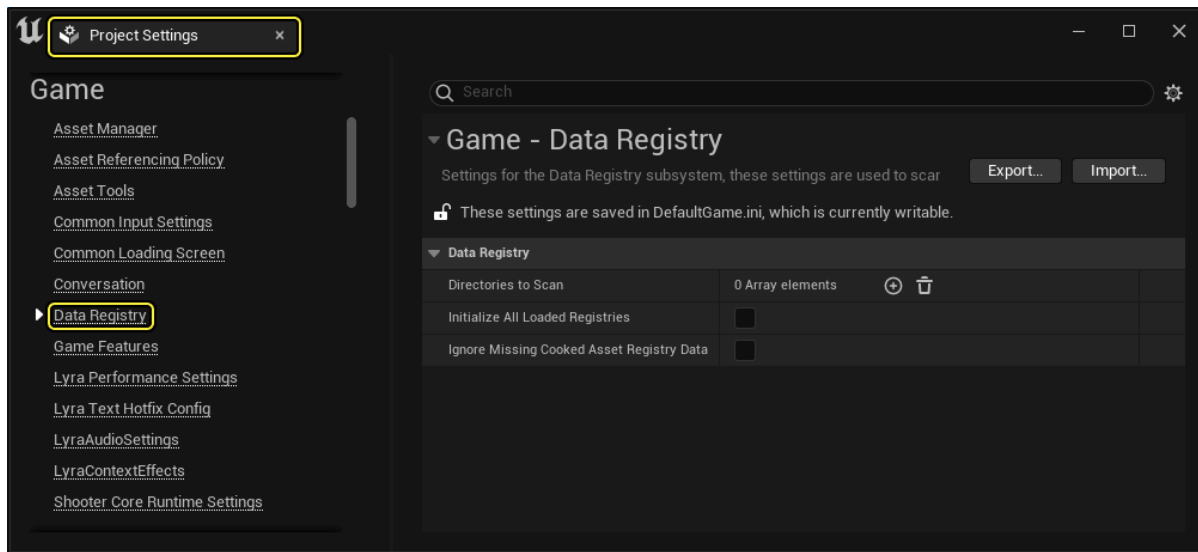
2. Navigate the **Project Settings** window by choosing **Edit > Project Settings**



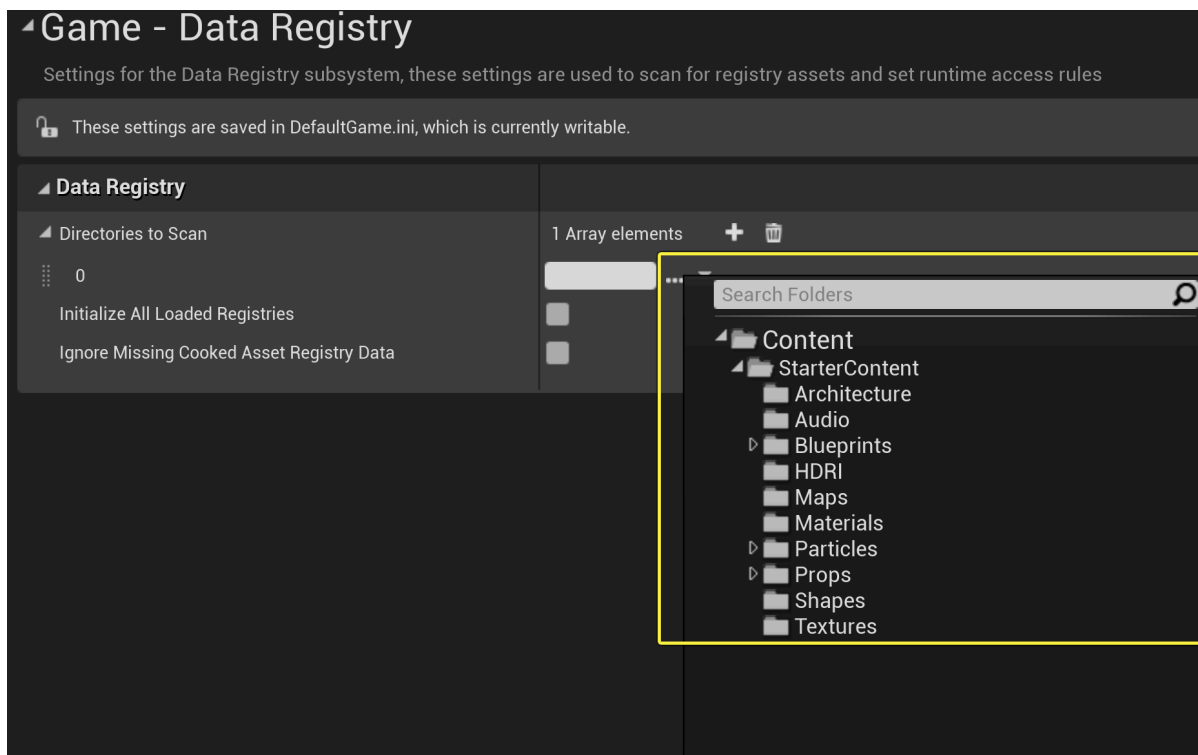
3. Under the **Game** section in the left panel, you will see a category labeled **Data Registry**.



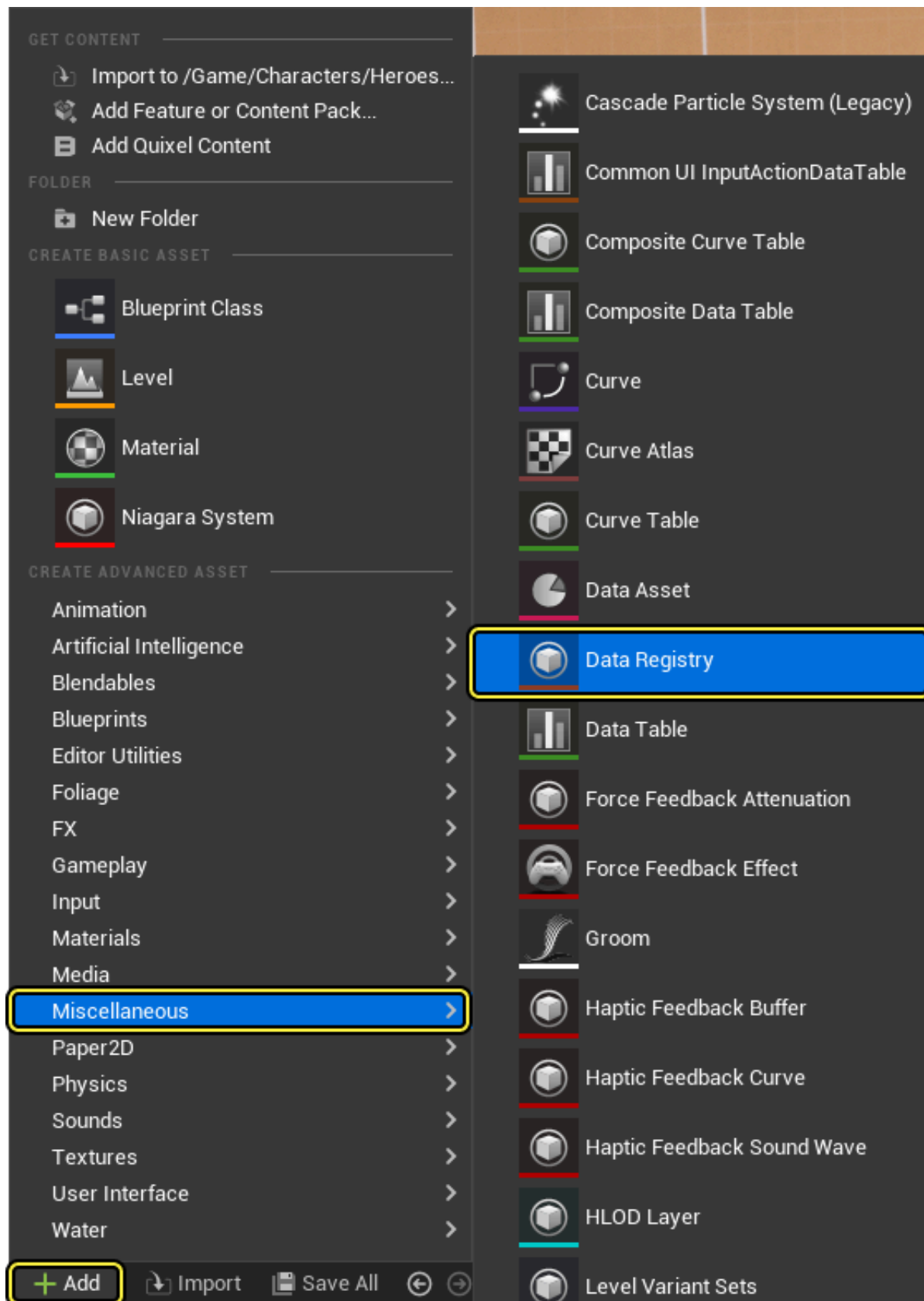
If you do not see this category, ensure that you have enabled the **Data Registry** plugin, and have restarted the editor at least once since enabling it.



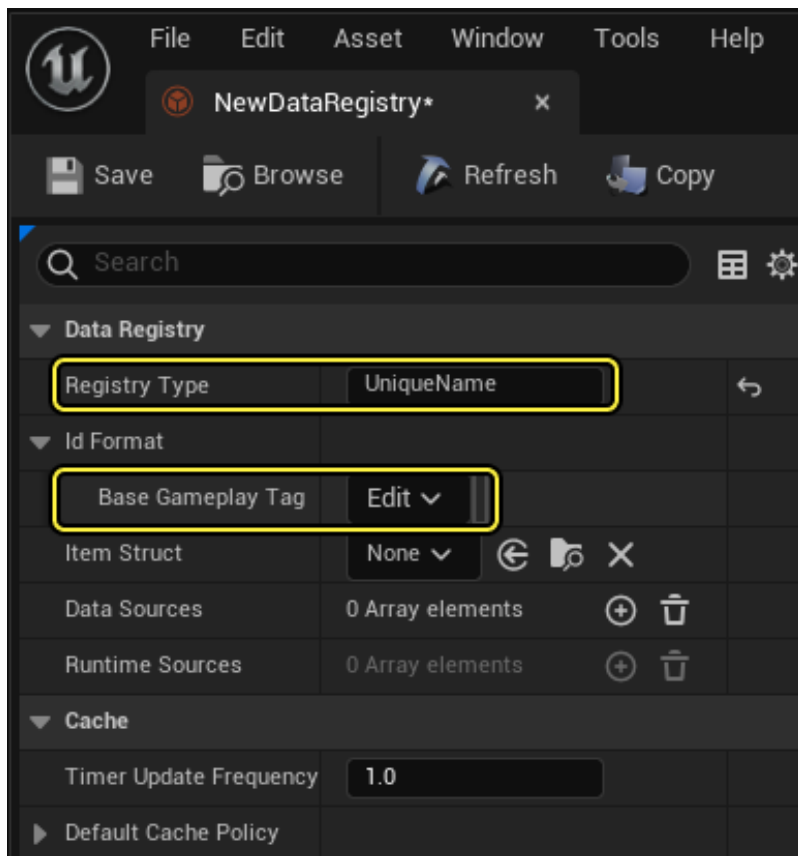
4. In the right-side panel, add at least one entry to the **Directories to Scan** array. This tells the system where to look to find Data Registry Assets. A typical entry might be `/Game/DataRegistries`, but you can add as many directories as you like, depending on the organizational scheme you prefer for your project. This is a good time to make sure that all of the directories you specified exist, and verify that any Data Registry Assets you have created are in the correct places.



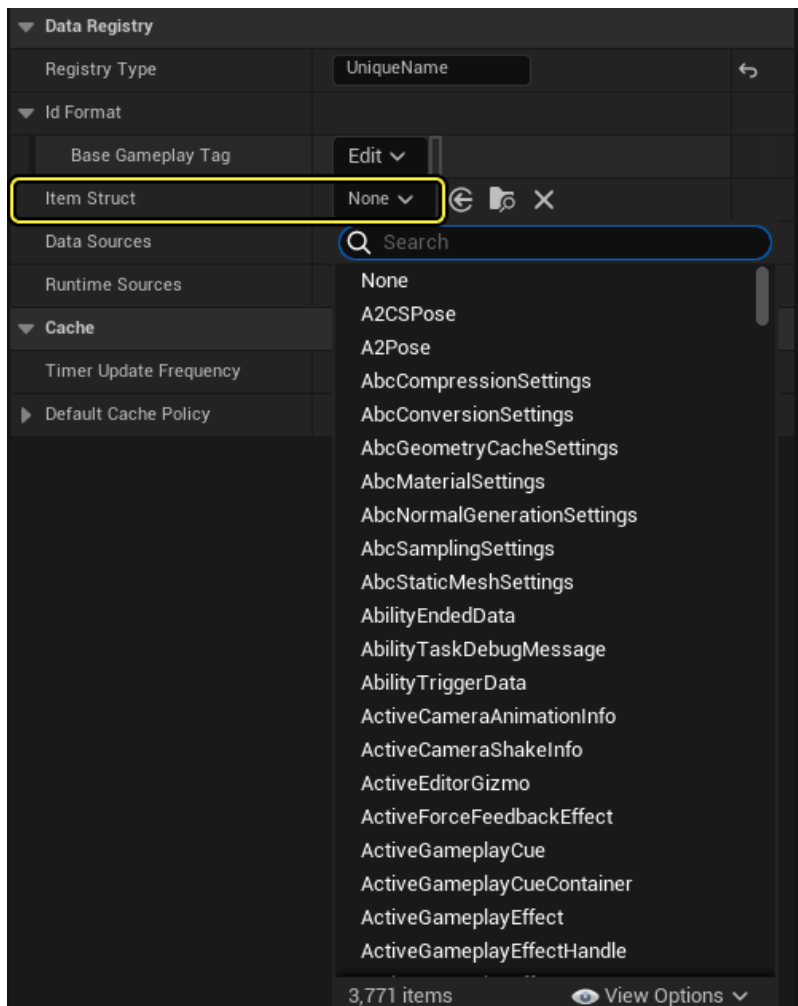
5. Navigate to one of the directories you specified in **Directories to Scan** in the Content Browser and add a new Data Registry Asset. You can do this by right-clicking an empty space in the right-side panel and expanding **Miscellaneous** in the context menu. In the expanded context menu, select **DataRegistry** and then pick the appropriate subclass from the list that appears.



6. Name and open your new Data Registry Asset.
7. Enter a globally unique name in the **Registry Type** field. If you want to use [Gameplay Tags](#) to identify Assets that should be part of this Data Registry, make sure to set it in the **Base Gameplay Tag** field under the **ID Format** category.



8. Set the **Item Struct** field to the type of struct that this Data Registry will contain. This is usually a **DataTableRowHandle** or a **SimpleCurve**, but can be any of a wide variety of data types.



9. Add one or more data sources to the **Data Sources** array. For each entry, you must choose what type of data you are adding, and specify where the data comes from. You can use the built-in data types, or add types that your project or any enabled plugin defines. At this point, you will need to have some data for the Data Registry to manage.
 - For simple source types like **DataTable Source** or **CurveTable Source**, which gather row data from known, pre-existing Assets, select the source Asset to read. The **Table Rules** section describes the caching behavior for data rows coming from the Asset you specified, so check that they are appropriate for your use case.
 - For meta source types like **DataTable Meta Source** or **CurveTable Meta Source**, which generate new data sources at runtime, there are a few more settings to handle. Select the source type to generate, any access or reference rules for the generated sources, and the scan or usage rules that determine how the system will find Assets. Meta sources can scan for Assets, accept registered Assets (from C++ code), or both.
 - As an example, you could specify the search path as `/Game/SearchableDataTables`, but exclude `*PrivateData*`; this exclusion would remove any path that contains the substring "PrivateData", so paths like `/Game/SearchableDataTables/PrivateData/`

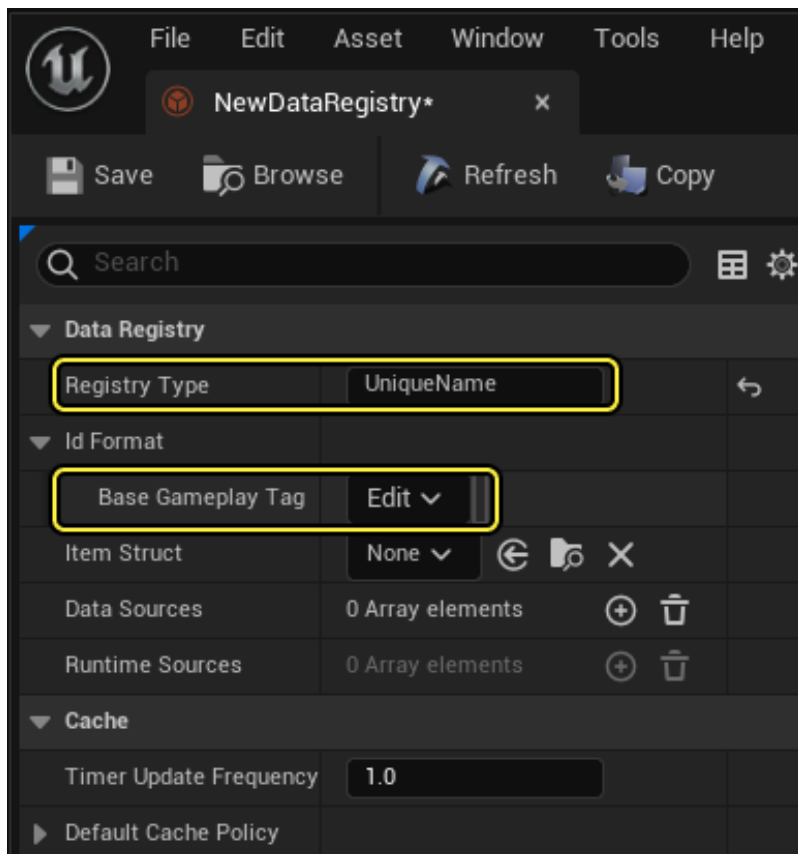
and `/Game/SearchableDataTables/SubPath/SomePrivateDataHere/` would not be scanned.

10. Once you have set up your data sources, configure the **Cache Rules** to meet the specific needs of your project. To find the settings that you can configure, expand the **Cache** section in the Data Registry editor.



Because the Data Registry feature is still in development, some of the settings might not work as expected.

11. This is a good time to ensure that the Data Registry contains the items that you expect. Click the **Refresh** button on the toolbar and review the contents of the **Registry Preview** tab, which shows a preview of every item currently known to the Data Registry. If the results do not match your expectation, modify the data source rules and click the **Refresh** button again.



Rows can appear multiple times in the list if multiple sources pull them in; this is not an error, and generally indicates that a meta source has found the same data as a simple source or another meta source. The order matters when you attempt to retrieve a data

item that appears in multiple sources. Only the first instance that the Data Registry discovered will be accessible.



You can Browse the **Runtime Sources** category, below **Data Sources** in the left panel, to see the source of each instance of each item in the Data Registry, listed in load order.

12. Once everything looks correct, save your Data Registry. In future sessions, it will load and populate itself automatically during startup. If you make changes to your Data Registry, press the Refresh button to update the data items.
13. If you plan to use Data Registries from C++ code, open your project's `Build.cs` file. The actual file name will include the name of your project; for example, if your project's name is "MyProject", the file name will be `MyProject.Build.cs`. Find the line that sets up the `PublicDependencyModuleNames` variable, and add `"DataRegistry"` to the array. The final line should look something like this:

```
PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreU"
```

Copy full snippet

After completing these steps, your project will be set up to use Data Registries! For information on how Data Registries work and what benefits they offer, refer to the main [Data Registries](#) page.