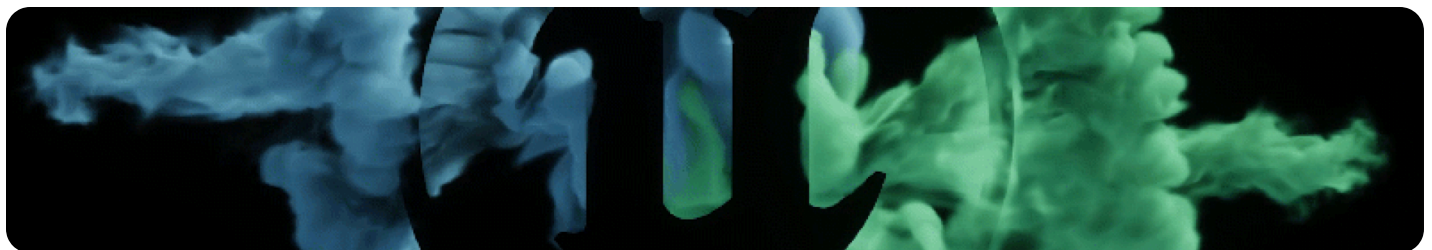


Niagara Data Channels Overview

Overview of Data Channels in Niagara



Overview

Niagara Data Channels (NDCs) facilitate communication between game code and Niagara Systems, or between different Niagara Systems.

Data channels are a stream of data with a defined payload that game code or Niagara Systems can read from or write to. Niagara Systems can read the payload and modify their behavior based on that information. They can also write information to a data channel, and that information can be used by other Niagara Systems or Blueprints during gameplay. A project can have multiple specialized data channel types for a variety of purposes.

One common use case for Data Channels is Niagara impact effects, where the player may spawn the same Niagara System multiple times during gameplay. Each system spawns and executes individually. This can become expensive if the player spawns many such systems rapidly.

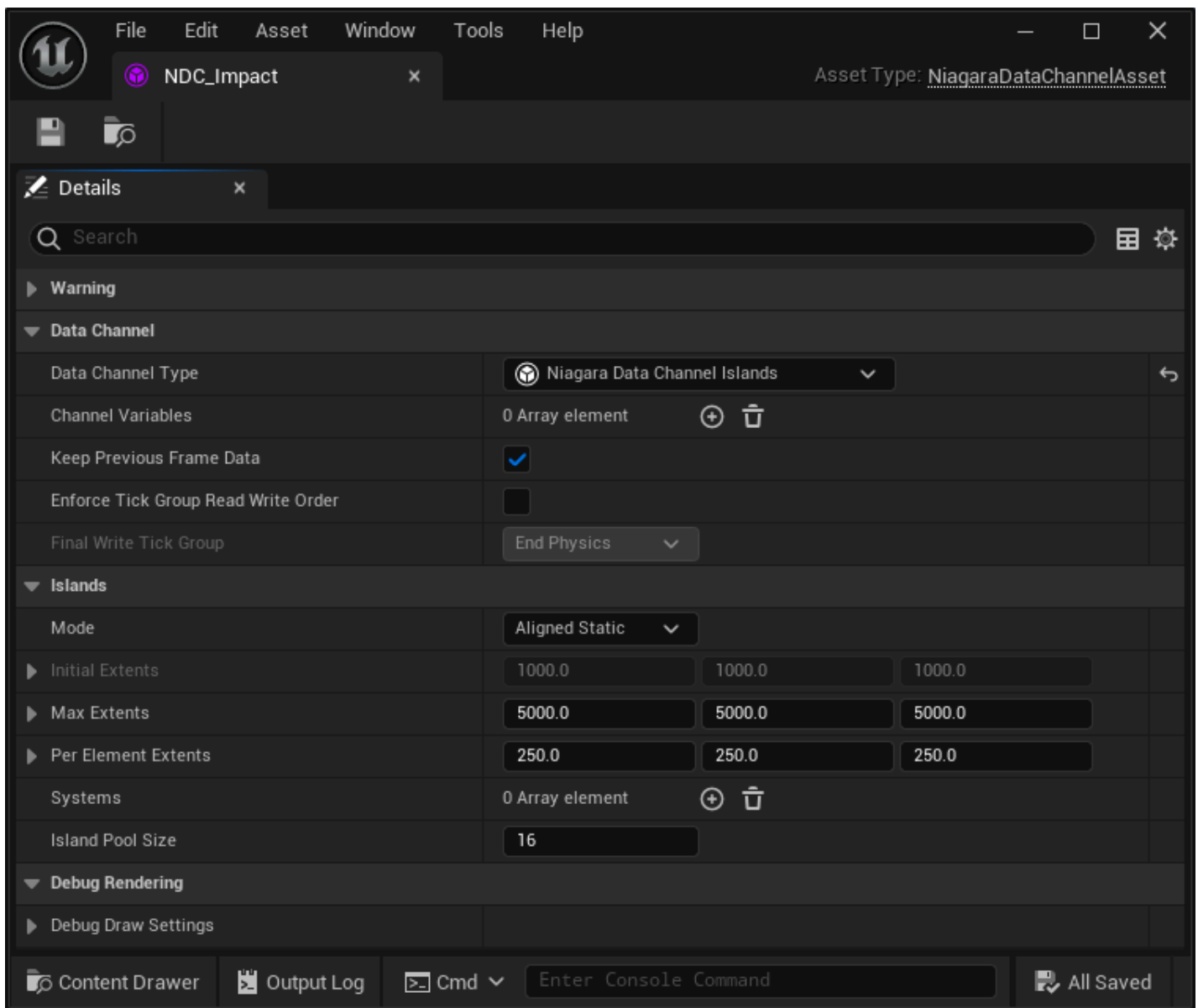
Niagara Data Channels provides an alternative where you can optimize burst Niagara Systems by combining them into a large shared simulation. So instead of spawning multiple separate Niagara Systems, you spawn a single system that handles all burst particles assigned to the data channel. This can provide significant performance gains.

Key Classes and Concepts

Using Data Channels with Niagara requires the following key components:

- A **Data Channel Asset** that holds the variables (payload) that will be written into the channel, along with a list of Niagara Systems that will use the channel.
- A **Niagara System** configured to be a continuous listener system. One of these Niagara Systems will be spawned per island and it will listen to data channel events. The system can use this information to spawn particles.
- A **Blueprint** that writes into the Data Channel and passes the specified information (payload).

Niagara Data Channel Asset

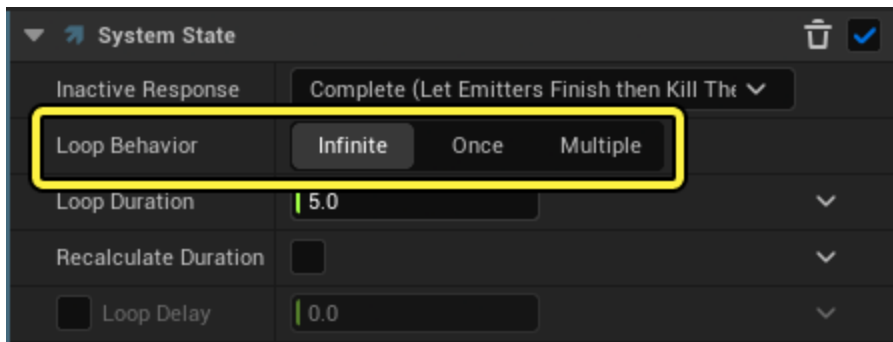


This asset holds common data channel settings, such as the data channel type and channel variables. If using the islands type, you can specify the initial and max extents of the islands and the island pool size. The variables can be common types, such as Float or Vector 4, as well as enumerators such as collision channel, physical surface, or Niagara Execution State. You can also add variables that represent specific Chaos destruction and Niagara collision events.

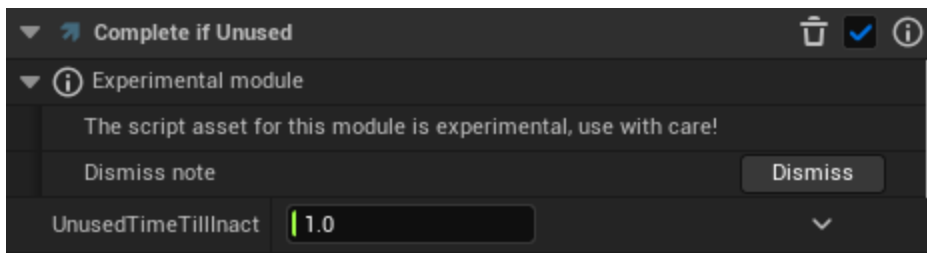
Each Data Channel contains a list of Niagara Systems that can be spawned within island bounds when an event is submitted to the Data Channel.

Niagara System

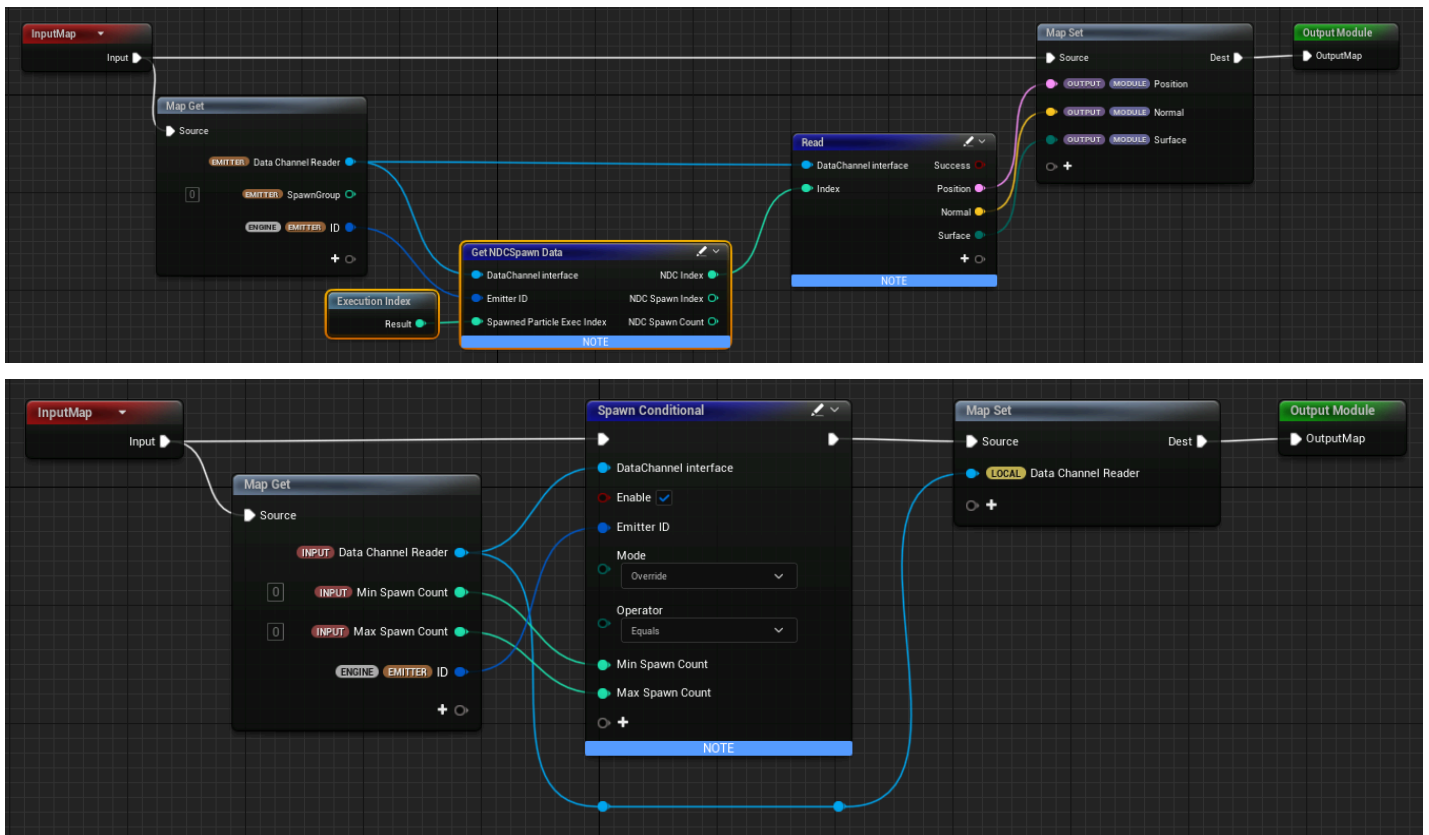
Your Niagara System will listen for events in the data channel and use that information to spawn particles. This system must be configured with an Infinite Loop Behavior so once it spawns in the island it can stay active as long as there are events being received from the data channel.



The system should also have a Complete if Unused module. This will destroy the system once there are no more events received after a certain amount of time. This will also allow the data channel to clean the island once there are no more Niagara Systems remaining.

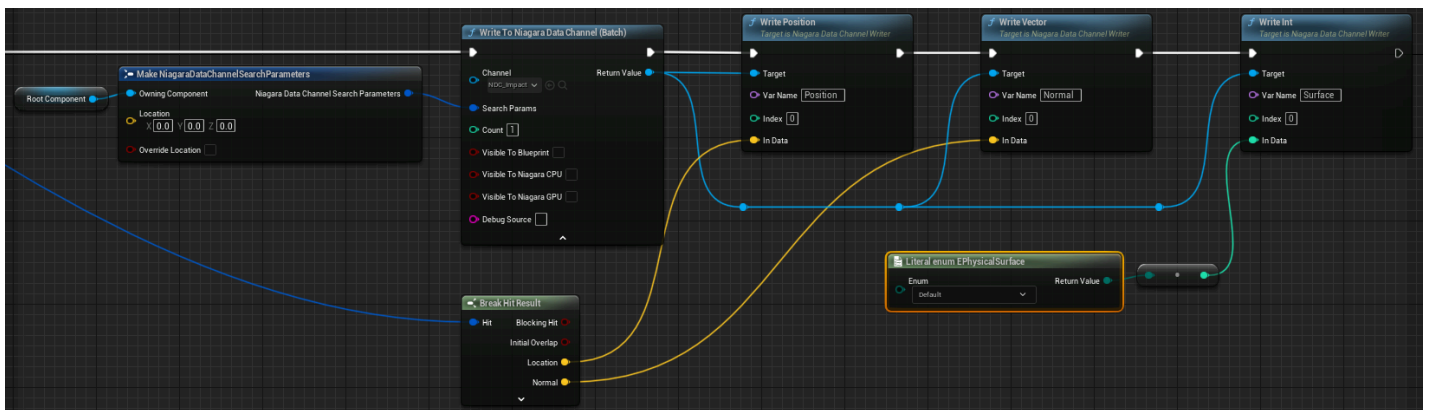


The emitter should have two Scratchpad modules - one for reading data from the Data Channel and one for spawning particles from the data it read from the data. channel.



Blueprint

Your Blueprint will write directly into the Niagara Data Channel and set the relevant variables.



For a step by step guide on how to use Niagara Data Channels, take a look at the [Niagara Data Channels Intro](#) in the EDC.