

Custom Java Datastore

[Here is a simple example](#) that dumps the output from the datastore to the console.

The documentation that describes the interfaces is shown in the [Data Designer manual Chapter 10 Object DataSource -> Object DataSource API -> DataListener](#) and there are some comments shown in the source code as well.

A very common question which was asked about the Custom Java DataStore is : Is the custom Java DataStore thread safe?

Thread safety is an issue when multiple threads have access to shared resources. In the case of our DataStores, each DataStore instance is only accessible by one thread, so thread safety is not an issue.

However, let us take a custom Java DataStore as an example:

There is one class MyDS.class which is my custom datastore.

Case 1: The user runs two separate composites, each of which reference MyDS.

Case 2: The user runs the same composite twice

In each case we will assume that both runs are in parallel – the user creates two threads and calls GenerateData in both of them.

Case 1: Each composite (running in a separate thread) will create an instance of MyDS

Case 2: Each composite instance will create an instance of MyDS

In both cases we are not concerned about thread safety because there is no sharing of instances between threads.

In general: every call to GenerateData will create an instance of the Composite which will create an instance of MyDS. No sharing.

However if the writer of MyDS is not aware of thread safety, they might use static variables in MyDS which will be shared. It is possible to write a custom DataStore that is not thread safe.