

Custom DataSets

Module Objectives

In this module, you will look at:

- What is a Custom DataSet?
 - Declaring a Custom DataSet
 - How to use Custom DataSets
-

Custom DataSets

Your own DataSet classes that implement common data patterns specific to your code

Suppose you want to define a counter table that, in addition to counting words, counts how many unique words it has seen

The DataSet can be built on top two underlying DataSets

- a first Table (entryCountTable) to count all the words and
- a second Table (uniqueCountTable) for the unique count

```
public class UniqueCountTable extends DataSet {  
  
    private Table entryCountTable;  
    private Table uniqueCountTable;  
}
```

Configuration and Initialization methods

Custom DataSets can also optionally implement

A **configure** method:

- Returns a specification which can be used to save metadata about the DataSet (such as configuration parameters)

An **initialize** method:

- Called at execution time
 - Any operations on the data of this DataSet are prohibited in `initialize()`
-

Custom DataSets Example (1 of 3)

Now we can begin with the implementation of the UniqueCountTable logic

Start with a few constants:

```
// Column name used for storing count of each entry.  
private static final byte[] ENTRY_COUNT = Bytes.toBytes("count");  
  
// Row and column name used for storing the unique count.  
private static final byte [] UNIQUE_COUNT = Bytes.toBytes("unique");
```

Custom DataSets Example (2 of 3)

- `UniqueCountTable` stores a counter for each word in its own row of the entry count table
- For each word the counter is incremented
- If the result of the increment is 1, then:
 - this is the first time the word has been encountered, hence a new unique word
 - increment the unique counter by 1:

```
public void updateUniqueCount(String entry) {  
    long newCount = entryCountTable.increment(Bytes.toBytes(entry), ENTRY_COUNT, 1L);  
    if (newCount == 1L) {  
        uniqueCountTable.increment(UNIQUE_COUNT, UNIQUE_COUNT, 1L);  
    }  
}
```

Custom DataSets Example (3 of 3)

Finally, write a method to retrieve the number of unique words seen:

```
public Long readUniqueCount() {  
    return uniqueCountTable.get(new Get(UNIQUE_COUNT, UNIQUE_COUNT))  
        .getLong(UNIQUE_COUNT, 0);  
}
```

Module Summary

You should be able to:

- Describe what are Custom DataSets
 - Design a Custom DataSet
 - Use Custom DataSets in an Application
-

Module Completed