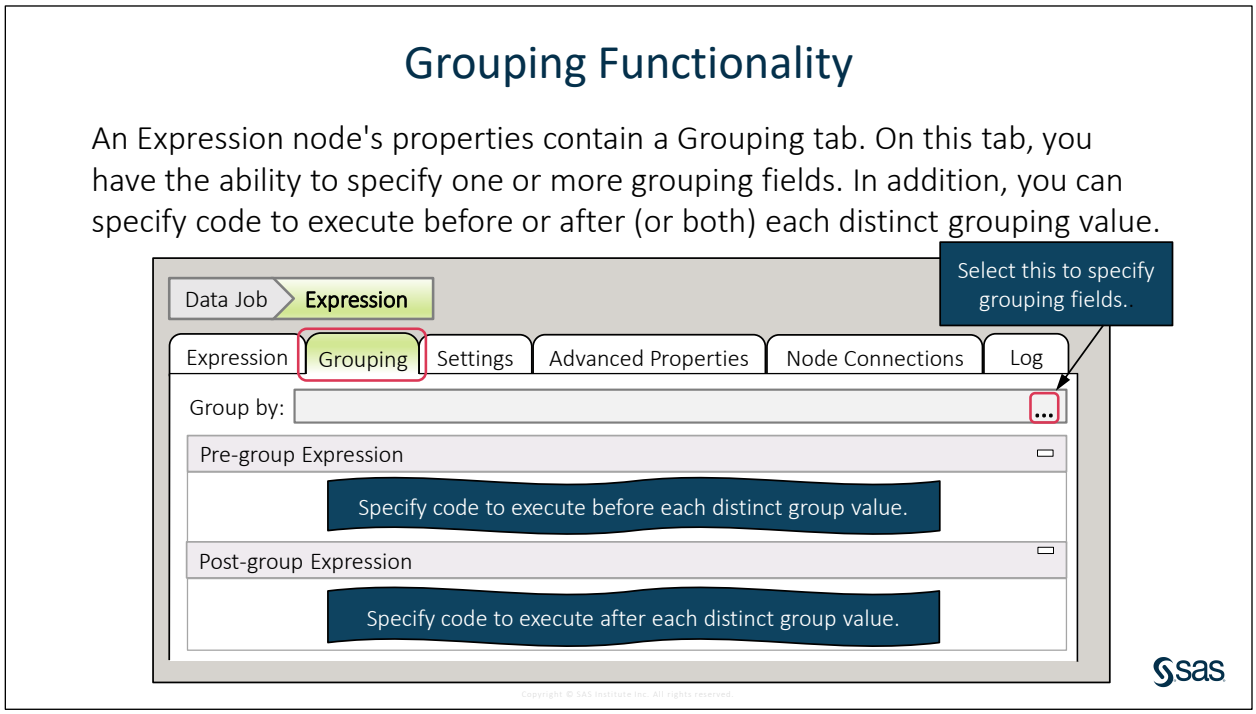
Using Grouping Functionality in Expression Node in Data Job

Grouping Functionality



We have seen the Expression tab in the Expression node.

An Expression node (in a data job) also has a Grouping tab.

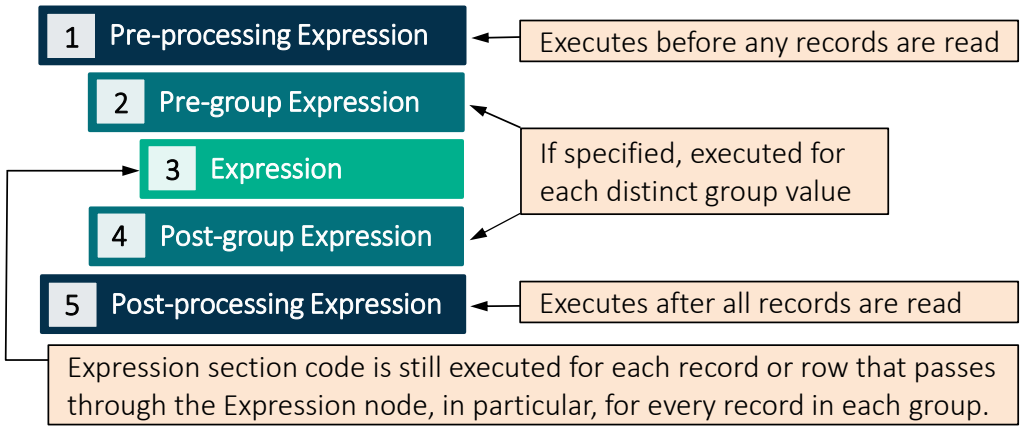
The Grouping tab

- provides the ability to specify one or more grouping fields
- provides two places for additional expression code
  - a Pre-group Expression area
  - a Post-group Expression area.

The Pre-group Expression area is to be used for code to execute before each distinct grouping value. The Post-group Expression area is to be used for code to execute after each distinct grouping value.

Processing Order with Grouping Code

The Grouping tab provides the ability to enter code in two more places (in addition to the three areas of the Expression tab). Here is the processing order of all *five* areas:



The order of processing across the five possible sections of expression code is as follows:

1. **Pre-processing Expression** – Code in the Pre-processing Expression area is processed before any records are read. This is an ideal place to declare and initialize variables.
2. **Pre-group Expression** – Code in the Pre-group Expression area is processed at the beginning of each new grouping.
3. **Expression** – Code in the Expression area is processed once for each record.
4. **Post-group Expression** – Code in the Post-group Expression area is processed at the end of each grouping.
5. **Post-processing Expression** – Code in the Post-processing Expression area is processed after every record is read. This is an appropriate place to assign final values to variables before the output is created.

Setup for the Questions

Consider the following set of data:

Group	X	Y
A	1	11
A	2	12
B	1	13
B	2	14
B	3	15
C	1	16
C	2	16
D	1	17

Number of Observations = 10

12

sas

x.01 Multiple Choice Question

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Pre-processing Expression section execute?

a. once
b. four times
c. ten times
d. never

13

sas

x.02 Multiple Choice Question

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Pre-group Expression section execute?

a. once
b. four times
c. ten times
d. never

15

x.03 Multiple Choice Question

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Post-group Expression section execute?

a. once
b. four times
c. ten times
d. never

17

x.04 Multiple Choice Question

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

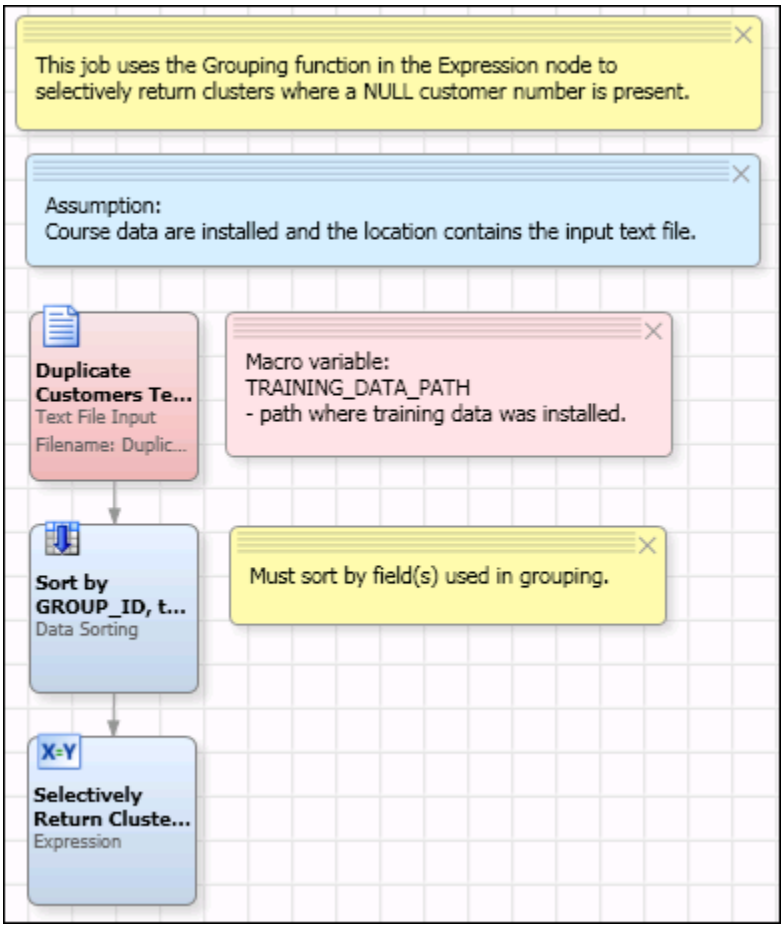
How many times does the Expression section execute?

a. once
b. four times
c. ten times
d. never

Using the Grouping Functionality

This demonstration illustrates a data job that uses the grouping functionality in an Expression node. This presents an alternative way of achieving the same result set from a previous demonstration.

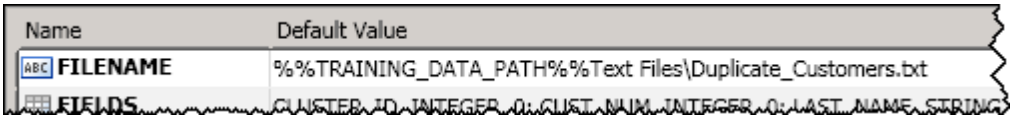
1. If necessary, invoke Data Management Studio.
   a. Select **Start** ⇨ **All Programs** ⇨ **DataFlux** ⇨ **Data Management Studio 2.7**.
   b. Click **Cancel** in the Log On window.
2. Access an existing data job.
   a. Click the **Folders** riser bar.
   b. Expand the **Advanced Demos** repository.
   c. Click the **batch_jobs** folder.
   d. Double-click the **Ch10D4_Grouping_Example** data job.



3. Review the properties of the Text File Input node.
   a. Right-click the node that is labeled **Duplicate Customers Text File** and select **Properties**.

      A window appears and states that macros are in use.

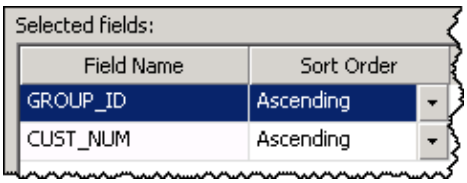
   b. Click **Open standard dialog**.
   c. Verify that the selected file is **D:\Workshop\dqdmp2\data\Text Files\Duplicate_Customers.txt**.
   d. Verify that the text qualifier is a double quotation mark (“) and the field delimiter is a comma. Also verify that the text file contains a header record.
   e. Verify that four fields are defined.
   f. Click **Cancel** to close the Text File Input Properties window.
4. Review the advanced properties of the Text File Input node.
   a. Right-click the node that is labeled **Duplicate Customers Text File** and select **Advanced Properties**.
   b. Verify that the **TRAINING_DATA_PATH** macro is used in the FILENAME attribute.



      Note: Recall the following assignment that is established in a .configuration (.cfg) file:

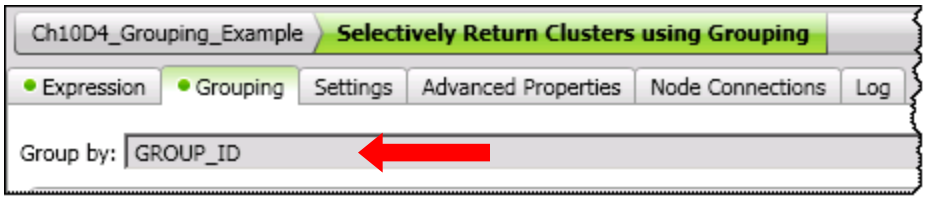
      TRAINING_DATA_PATH = D:\Workshop\dqdmp2\data\

   c. Click **Cancel** to close the Advanced Properties window.
5. Review the properties of the Data Sorting node.
   a. Right-click the node that is labeled **Sort by GROUP_ID, then by Cust_Num** and select **Properties**.
   b. Verify that the **GROUP_ID** and **CUST_NUM** fields are selected for sorting.



   c. Click **Cancel** to close the Data Sorting Properties window.
6. Review the properties of the Expression node.
   a. Right-click the node that is labeled **Selectively Return Groups using Grouping** and select **Properties**.
   b. Click the **Grouping** tab.

1) Verify that **GROUP_ID** is selected for the **Group by** field.



2) Verify that the following code is specified in the Pre-group Expression section:

```
// variable to indicate first record in the group
hidden boolean first_rec
first_rec = true
```

3) Verify that no code is specified for the Post-group Expression section. (The absence of a green dot is a visual indicator.)

c. Click the **Expression** tab.

d. Verify that the following code is specified in the Pre-processing Expression section:

```
// variable to indicate whether or not to keep record
hidden boolean keep
```

e. Verify that the following code is specified in the Expression section:

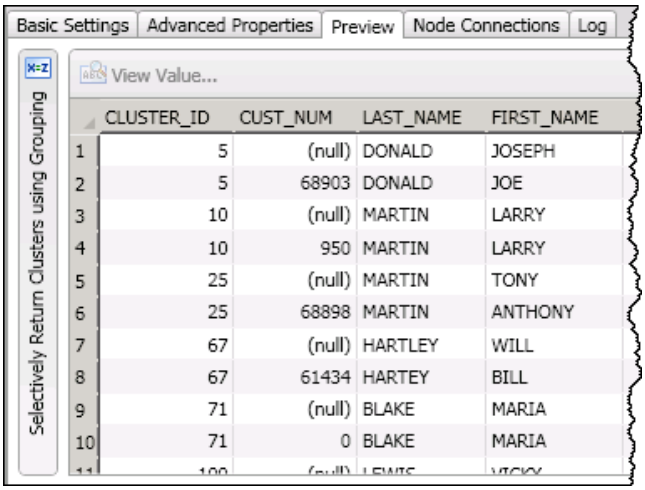
```
/* Records must be pre-sorted by GROUP_ID and Cust_Num.
   If Cust_Num is null then this will be first record
   in the group. */
if first_rec
   begin
     if isnull(`CUST_NUM`)
          keep = true
     else keep = false
     first_rec = false
   end
/* can comment out this section to see values for keep -
   will need to unhide variable in pre-processing.        */
if keep
   return true
else return false
```

f. Click the main item on the thread to return to the Data Flow tab.

7. Preview the Expression node.

a. If necessary, select **View** ⇨ **Show Details Pane**.

b. Right-click the **Expression** node and select **Preview**.



	CLUSTER_ID	CUST_NUM	LAST_NAME	FIRST_NAME
1	5	(null)	DONALD	JOSEPH
2	5	68903	DONALD	JOE
3	10	(null)	MARTIN	LARRY
4	10	950	MARTIN	LARRY
5	25	(null)	MARTIN	TONY
6	25	68898	MARTIN	ANTHONY
7	67	(null)	HARTLEY	WILL
8	67	61434	HARTEY	BILL
9	71	(null)	BLAKE	MARIA
10	71	0	BLAKE	MARIA

8. Change the code to view the **keep** field.
   a. Right-click the node that is labeled **Selectively Return Groups using Grouping** and select **Properties**.
   b. Verify that the **Expression** tab is selected.
   c. In the Pre-processing Expression section, remove the keyword **hidden** from the declaration statement for the **keep** field.

```
hidden boolean keep
```

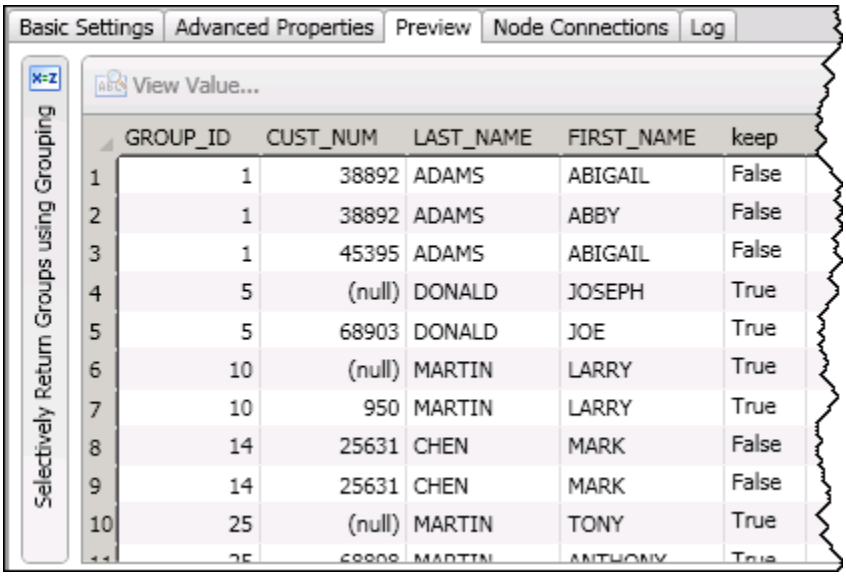
```
boolean keep
```

   d. In the Expression section, comment out the last portion of the code to see the values for the **keep** field.

```
if keep
   return true
else return false
```

```
/*
if keep
   return true
else return false
*/
```

   e. Click the main item on the thread to return to the Data Flow tab.
9. Select **File** ⇨ **Save**.
10. Preview the Expression node.
    a. If necessary, select **View** ⇨ **Show Details Pane**.
    b. Right-click the **Expression** node and select **Preview**.



	GROUP_ID	CUST_NUM	LAST_NAME	FIRST_NAME	keep
1	1	38892	ADAMS	ABIGAIL	False
2	1	38892	ADAMS	ABBY	False
3	1	45395	ADAMS	ABIGAIL	False
4	5	(null)	DONALD	JOSEPH	True
5	5	68903	DONALD	JOE	True
6	10	(null)	MARTIN	LARRY	True
7	10	950	MARTIN	LARRY	True
8	14	25631	CHEN	MARK	False
9	14	25631	CHEN	MARK	False
10	25	(null)	MARTIN	TONY	True

Each record that is marked *False* is a record for a group in which there are no *(null)* values for **CUST_NUM**.

11. Change the code back to the original code where the **keep** field is hidden.
    a. Right-click the node labeled **Selectively Return Groups using Grouping** and select **Properties**.
    b. Verify that the **Expression** tab is selected.
    c. In the Pre-processing Expression section, add the keyword **hidden** to the declaration statement for the **keep** field.

```
boolean keep
```

```
hidden boolean keep
```

    d. In the Expression section, uncomment the last portion of the code.

```
/*
if keep
   return true
else return false
*/
```

```
if keep
   return true
else return false
```

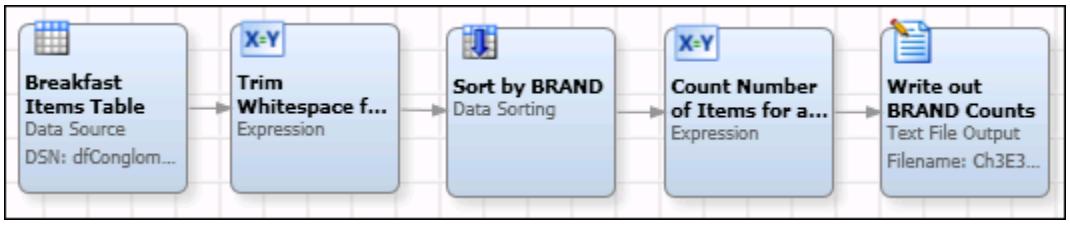
    e. Click the main item on the thread (labeled **Ch10D4_Grouping_Example**) to return to the Data Flow tab.
12. Select **File** ⇨ **Save**.
13. Select **File** ⇨ **Close**.

As mentioned previously, a final job flow would likely contain additional nodes to process the output from the Expression node. Because you are investigating only the grouping functionality, and this was accomplished by previewing the node, do not run this data job. **End of Demonstration**

Exercise

1. Using the Expression Node with the Grouping Functionality

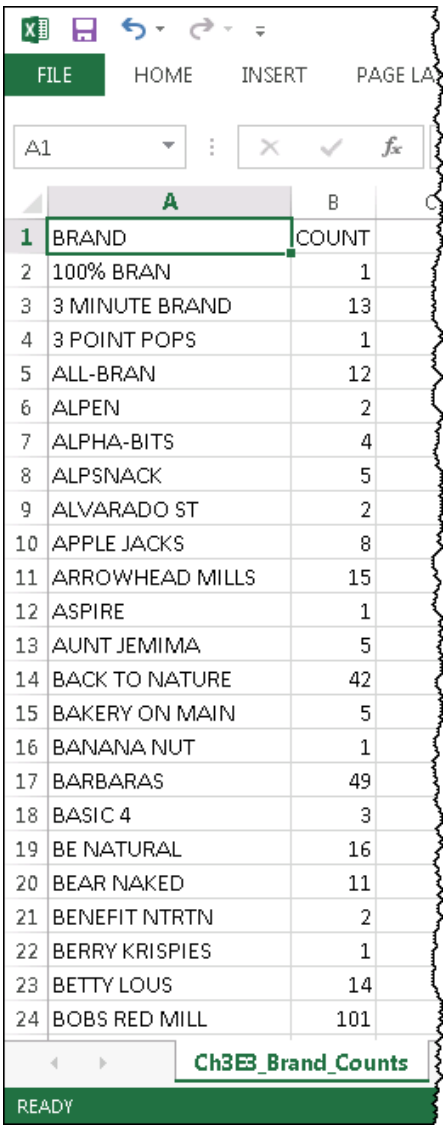
Create a data job that counts the number of items within a brand. Use the grouping functionality in the Expression node. Write only the summary information for each brand including its total count of items to a text file.



Use the following specifications:

- Create the data job in the **batch_jobs** folder of the **Advanced Exercises** repository. Enter **Ch10E3_Grouping_Exercise** as the name.
- Use the **BREAKFAST_ITEMS** table (from the **dfConglomerate Grocery** data connection). Select only the **BRAND** field from this table.
- Use an Expression node to trim the whitespace characters from the **BRAND** field values. Hint: Use the TRIM function.
- Use the Data Sorting node to sort the data by **BRAND**.
- Use a second Expression node with the Grouping functionality to calculate the count of each brand value.
  - For pre-processing, declare a counter field as an integer.
  - Specify the group information:
    - Select the **BRAND** field.
    - Using the Pre-group Expression section, initialize the declared counter to 0.
    - Using the Post-group Expression section, push the summary row to the stack for output.
  - For the Expression section of the Expression tab, increase the counter by 1. Also, specify that the individual rows should ***not*** be returned. (Use “return false” as a statement.)
- Use a Text File Output node to write the results to a text file.
  - Specify **D:\Workshop\dqdmp2\FTExercises\files\output_files\Ch10E3_BrandCounts.csv** as the value for the **File name** attribute.
  - Use a double quotation mark (“) as the text qualifier and a comma as the delimiter. Include a header record and display the file after the job runs.
- Save and run the data job.

The output file should resemble the following:



	A	B
1	BRAND	COUNT
2	100% BRAN	1
3	3 MINUTE BRAND	13
4	3 POINT POPS	1
5	ALL-BRAN	12
6	ALPEN	2
7	ALPHA-BITS	4
8	ALPSNACK	5
9	ALVARADO ST	2
10	APPLE JACKS	8
11	ARROWHEAD MILLS	15
12	ASPIRE	1
13	AUNT JEMIMA	5
14	BACK TO NATURE	42
15	BAKERY ON MAIN	5
16	BANANA NUT	1
17	BARBARAS	49
18	BASIC 4	3
19	BE NATURAL	16
20	BEAR NAKED	11
21	BENEFIT NTRTN	2
22	BERRY KRISPIES	1
23	BETTY LOUS	14
24	BOBS RED MILL	101

Question: How many records were read from the BREAKFAST_ITEMS table?

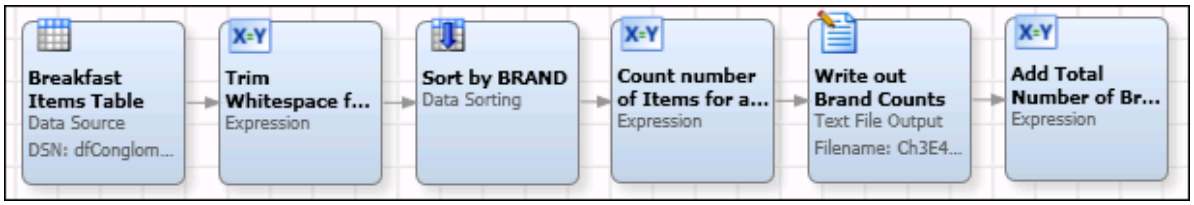
Answer: _____

Question: How many records were written to the file?

Answer: _____

2. (Optional) Using the Expression Node with the Grouping Functionality

Create a copy of the data job from the previous exercise. This copy accumulates the total number of brands and adds this information to the output file that is created. This can be accomplished by adding an additional Expression node to the existing job flow.



Use the following specifications:

- Place the copied data job in the **batch_jobs** folder of the **Advanced Exercises** repository and rename it as **Ch10E4_Grouping_Exercise_Extra**.
- Modify the Text File Output node to create an output file named **Ch10E4_Brand_Counts_Extra.csv**.
- Add an additional Expression node that does the following:
  - declares a counter field in pre-processing.
  - accumulates the counter in expression code.
  - uses post-processing to write two blank lines and a message to the same file that is specified in the Text File Output node. The message should be something similar to the following: TOTAL NUMBER OF BRANDS: <*value*>.

    Hint: Use the file object functions OPEN, SEEKEND, WRITELINE, and CLOSE.
- Save and run the data job.

Question: How many distinct brands exist?

Answer: _____

End of Exercises

Solutions

Solutions to Exercises

1. **Using the Expression Node with the Grouping Functionality**

   a. If necessary, invoke Data Management Studio.

   1) Select **Start** ⇨ **All Programs** ⇨ **DataFlux** ⇨ **Data Management Studio 2.7**.
   2) Click **Cancel** in the Log On window.

   b. Verify that the **Home** tab is selected.

   c. Create a new data job.

   1) Click the **Folders** riser bar.
   2) Expand the **Advanced Exercises** repository.
   3) Click the **batch_jobs** folder.
   4) Click  and select **Data Job**.
      a) Enter **Ch10E3_Grouping_Exercise** in the **Name** field.
      b) Click **OK**.

   d. Add the Data Source node to the job flow.

   1) Verify that the **Nodes** riser bar is selected in the resource pane.
   2) Expand the **Data Inputs** grouping of nodes.
   3) Double-click the **Data Source** node.
   4) Enter **Breakfast Items Table** in the **Name** field.
   5) Click the ellipsis  next to the **Input table** field.
      a) Expand the **dfConglomerate Grocery** data connection.
      b) Click **BREAKFAST_ITEMS**.
      c) Click **OK** to close the Select Table window.
   6) Select only the needed field.
      a) Click the left-pointing double arrow  to remove all selected fields from the Selected list.
      b) Double-click the **BRAND** field in the Available list to move it to the Selected list.
   7) Click **OK** to save the changes and close the Data Source Properties window.

      The Data Source node appears in the job diagram where the information is displayed.

   e. Add an Expression node to the job flow.

   1) Verify that the **Nodes** riser bar is selected in the resource pane.
   2) Collapse the **Data Inputs** grouping of nodes.
   3) Expand the **Utilities** grouping of nodes.

4) Double-click the **Expression** node.

   The Expression node is added to the data flow, and the job drills into the properties for the Expression node.

f. Specify properties for the Expression node.

1) Click the **Settings** tab.
2) Enter **Trim Whitespace from BRAND** in the **Name** field.
3) Enter **Expression** in the **Description** field.
4) Click the **Expression** tab.
5) Enter the following code in the Expression section:

```
BRAND = trim(BRAND)
```

6) Click the main item on the thread (labeled **Ch10E3_Grouping_Exercise**) to return to the Data Flow tab.

g. Add a Data Sorting node to the job flow.

1) Verify that the **Nodes** riser bar is selected in the resource pane.
2) Collapse the **Utilities** grouping of nodes.
3) Expand the **Data Integration** grouping of nodes.
4) Double-click the **Data Sorting** node.

   The Data Sorting node is added to the data flow, and the Data Sorting Properties window appears.

h. Specify properties for the Data Sorting node.

1) Enter **Sort by BRAND** in the **Name** field.
2) If necessary, double-click the **BRAND** field to move it from the Available list to the Selected list.
3) Click **OK** to close the Data Sorting Properties window.

i. Add a second Expression node to the job flow.

1) Verify that the **Nodes** riser bar is selected in the resource pane.
2) Collapse the **Data Integration** grouping of nodes.
3) Expand the **Utilities** grouping of nodes.
4) Double-click the **Expression** node.

   The Expression node is added to the data flow, and the job drills in to the properties for the Expression node.

j. Specify properties for the Expression node.

1) Click the **Settings** tab.
   a) Enter **Count Number of Items for a BRAND** in the **Name** field.
   b) Enter **Expression** in the **Description** field.
2) Click the **Expression** tab.
   a) Enter the following code in the Pre-processing Expression section:

```
integer COUNT
```

b) Enter the following code in the Expression section:

```
COUNT = COUNT + 1
return false
```

3) Click the **Grouping** tab.

a) Click the ellipsis ... next to the **Group by** field.

Note: The ellipsis ... button is on far right side of the field.

(1) Select **BRAND** in the Group By window.

(2) Click **OK** to close the Group By window.

b) Enter the following code in the Pre-group Expression section:

```
COUNT = 0
```

c) Enter the following code in the Post-group Expression section:

```
pushrow()
```

4) Click the main item on the thread (labeled **Ch10E3_Grouping_Exercise**) to return to the Data Flow tab.

k. Add a Text File Output node to the job flow.

1) Verify that the **Nodes** riser bar is selected in the resource pane.

2) Collapse the **Utilities** grouping of nodes.

3) Expand the **Data Outputs** grouping of nodes.

4) Double-click the **Text File Output** node.

l. Specify properties for the Text File Output node.

1) Enter **Write out BRAND Counts** in the **Name** field.

2) Specify the output file information.

a) Click the ellipsis ... next to the **Output file** field.

b) Navigate to **D:\Workshop\dqdmp2\FTExercises\files\output_files**.

c) Enter **Ch10E3_Brand_Counts.csv** in the **File name** field.

d) Click **Save**.

3) Specify attributes for the file.

a) Verify that the **Text qualifier** field is set to " (double quotation mark).

b) Verify that the **Field delimiter** field is set to **Comma**.

c) Click **Include header row**.

d) Click **Display file after job runs**.

4) Verify that both the **BRAND** and **COUNT** fields are selected.

5) Click **OK** to save the changes and close the Text File Output Properties window.

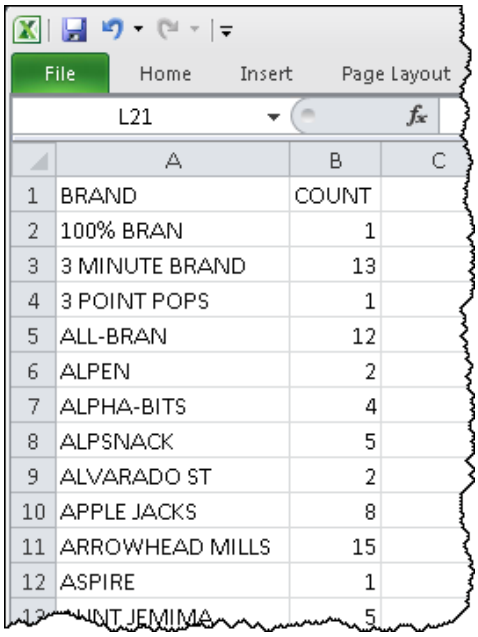
m. Save the job.

1) If necessary, click the **Data Flow** tab.
2) Select **File** ⇨ **Save**.

n. Run the job.

1) Verify that the **Data Flow** tab is selected.
2) Select **Actions** ⇨ **Run Data Job**.

The .csv file appears in Microsoft Excel:



	A	B	C
1	BRAND	COUNT	
2	100% BRAN	1	
3	3 MINUTE BRAND	13	
4	3 POINT POPS	1	
5	ALL-BRAN	12	
6	ALPEN	2	
7	ALPHA-BITS	4	
8	ALPSNACK	5	
9	ALVARADO ST	2	
10	APPLE JACKS	8	
11	ARROWHEAD MILLS	15	
12	ASPIRE	1	
13	AUNT JEMIMA	5	

3) After viewing the CSV file, select **File** ⇨ **Exit**. (If you are prompted, do ***not*** save changes.)

o. View the log information.

1) Click the **Log** tab.
2) Review the information for each of the nodes.

p. When you are finished reviewing the information, click the **Data Flow** tab.

q. Select **File** ⇨ **Close**.

2. (Optional) Using the Expression Node with the Grouping Functionality

a. Create a copy of the **Ch10E3_Grouping_Exercise** data job.

1) Click the **Folders** riser bar.
2) Expand the **Advanced Exercises** repository.
3) Expand the **batch_jobs** folder.
4) Right-click the data job **Ch10E3_Grouping_Exercise** and select **Copy**.
5) Right-click the **batch_jobs** folder and select **Paste**.
   a) In the Item Already Exists window, click **Rename the new item**.

b) Click **OK**.

6) Right-click the new data job that is named **Copy of Ch10E3_Grouping_Exercise**. Select **Rename**.

7) Enter **Ch10E4_Grouping_Exercise_Extra** and press Enter.

b. Double-click the new data job **Ch10E4_Grouping_Exercise_Extra** to open it for editing.

c. Edit the Text File Output node.

1) Right-click the **Text File Output** node and select **Properties**.

2) Change the file name to **Ch10E4_Brand_Counts_Extra.csv**.

3) Click **OK** to close the Text File Output Properties window.

d. Add an Expression node to the job flow.

1) In the Nodes resource pane, expand the **Utilities** grouping of nodes.

2) Double-click the **Expression** node.

The Expression node is added to the data flow, and the job drills in to the properties for the Expression node.

e. Specify properties for the Expression node.

1) Click the **Settings** tab.

2) Enter **Add Total Number of Brands to Text File** in the **Name** field.

3) Enter **Expression** in the **Description** field.

4) Click the **Expression** tab.

5) Enter the following code in the Pre-processing Expression section:

```
hidden integer NUM_BRANDS
NUM_BRANDS = 0
```

6) Enter the following code in the Expression section:

```
NUM_BRANDS = NUM_BRANDS + 1
```

7) Enter the following code in the Post-processing Expression section:

```
file f
f.open("D:\Workshop\dqdmp2\AdvExercises\files\output_files\
        Ch10E4_Brand_Counts_Extra.csv","rw")
f.seekend(0)
f.writeline('')
f.writeline('')
f.writeline('TOTAL NUMBER OF BRANDS: ' & NUM_BRANDS)
f.close()
```

8) Click the main item on the thread (labeled **Ch10E4_Grouping_Exercise_Extra**) to return to the Data Flow tab.

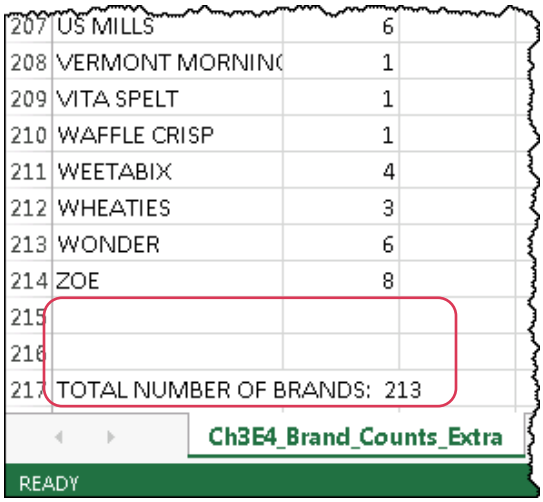
f. Save the job.

1) If necessary, click the **Data Flow** tab.

2) Select **File** ⇨ **Save**.

g. Run the job.

1) Verify that the **Data Flow** tab is selected.
2) Select **Actions** ⇨ **Run Data Job**.
3) When the CSV file appears, scroll to the end of the file.
4) Verify that the new text was appended to the file.



Question: How many distinct brands exist?

Answer: **213**

5) After you view the CSV file, select **File** ⇨ **Exit**. (If you are prompted, do ***not*** save the changes.)

h. View the log information.

1) Click the **Log** tab.
2) Review the information for each of the nodes.

i. When you are finished reviewing the information, select **File** ⇨ **Close**.

End of Solutions

Solutions to Activities and Questions

x.01 Multiple Choice Question – Correct Answer

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Pre-processing Expression section execute?

a. once
b. four times
c. ten times
d. never

The correct answer is a., because pre-processing expression code executes only once.

14

x.02 Multiple Choice Question – Correct Answer

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Pre-group Expression section execute?

a. once
b. four times
c. ten times
d. never

The correct answer is d., because no code is specified in the Pre-group Expression area. Therefore, there is nothing to execute.

16

x.03 Multiple Choice Question – Correct Answer

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Post-group Expression section execute?

a. once
b. four times
c. ten times
d. never

The correct answer is b., because there are four distinct values for the grouping field *Group*. Therefore, the post-group expression code executes after each distinct grouping value. That is, the post-group expression code executes four times.

18

sas

x.04 Multiple Choice Question – Correct Answer

An Expression node is added to a data job. Expression code is found in the Pre-processing Expression, Expression, and Post-group Expression sections. The field **Group** is defined as the **Group by** field.

How many times does the Expression section execute?

a. once
b. four times
c. ten times
d. never

The correct answer is c., because Expression section code executes once for each record or row that is processed. In this case, the code in the Expression section executes 10 times.

20

sas