

Expression Engine Language: Understanding PUSHROW Function

PUSHROW Function

PUSHROW Function

The PUSHROW function pushes the current values of all symbols to be written to output as a row or record. This includes both field values for the current row and the defined symbols in the code.



To help identify *pushed* rows, the Settings tab of an Expression node has the **Pushed status field** option.

Output

Generate rows when no parent is specified

Pushed status field: Pushed

Return status field:

Progress indicator: 10 green dots

The **Pushed status field** option specifies a field that displays the pushed status in the output.

107



By default, the pre-processing and post-processing sections do not interact with any rows of data. Thus, by default, you cannot use RETURN TRUE to pass a record out of the pre-processing or post-processing sections. The only option for writing out a record in either pre-processing or post-processing sections is using the PUSHROW function.

A pushed record or a set of records can be identified by using an option that defines a field that surfaces the pushed state.

The Settings tab of an Expression node has an option **Pushed status field** that specifies a field that displays the pushed status in the output. The field holds the value *true* when the row results from a PUSHROW function. Otherwise, *false* is displayed.

Demonstration: Post-processing (Old)

Recall that post-processing previously wrote calculated values to the log and a text file.

```
// create variables for percentage calculations
hidden real percent_late
hidden real percent_early
hidden real percent_ontime

// calculate percentages; round to 2 decimal places
percent_late = round(((late_count/count)*100),2)
percent_early = round(((early_count/count)*100),2)
percent_ontime = round(((ontime_count/count)*100),2)

// write percentage values to log
----CODE OMITTED----
//write percentage values to text file
----CODE OMITTED----
```

108



Copyright © 2019, SAS Institute Inc. All rights reserved.

Demonstration: Post-processing (New)

In this example, the calculated values are pushed to the stack and passed along as a row of data.

```
// create variables for percentage calculations
hidden real percent_late
hidden real percent_early
hidden real percent_ontime

// calculate percentages; round to 2 decimal places
percent_late = round(((late_count/count)*100),2)
percent_early = round(((early_count/count)*100),2)
percent_ontime = round(((ontime_count/count)*100),2)

// use pushrow() to push calculated values to stack
pushrow() ←
```

109



Copyright © 2019, SAS Institute Inc. All rights reserved.

Demonstration: Pushed Status Field

In addition, the **Pushed status field** option is investigated.

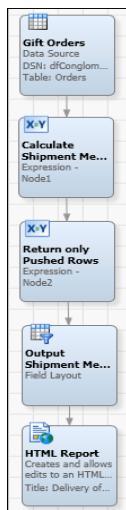
The screenshot shows the configuration window for a SAS node named 'Calculate Shipment Metrics'. The window has several tabs: 'Expression', 'Grouping', 'Settings', 'Advanced Properties', 'Node Connections', and 'Log'. The 'Expression' tab is active. Under the 'Identification' section, the 'Name' is 'Calculate Shipment Metrics' and the 'Description' is 'Expression - Node1'. The 'Notes' section is empty. Under the 'Output' section, there are three options: 'Generate rows when no parent is specified' (unchecked), 'Pushed status field: Pushed' (checked, with a red arrow pointing to the text), and 'Return status field:' (unchecked).

110

sas

Demonstration: Data Flow

The final data flow should resemble the following:



The Expression node has amended code that uses the **PUSHROW** function, and **Pushed status field** is selected. Also, the post-processing expression code no longer writes information to the log and a file.

The second Expression node "passes" records only where the **Pushed status field** option is set to *true*.

Additional nodes select and arrange only the desired fields for an HTML report.

111

sas

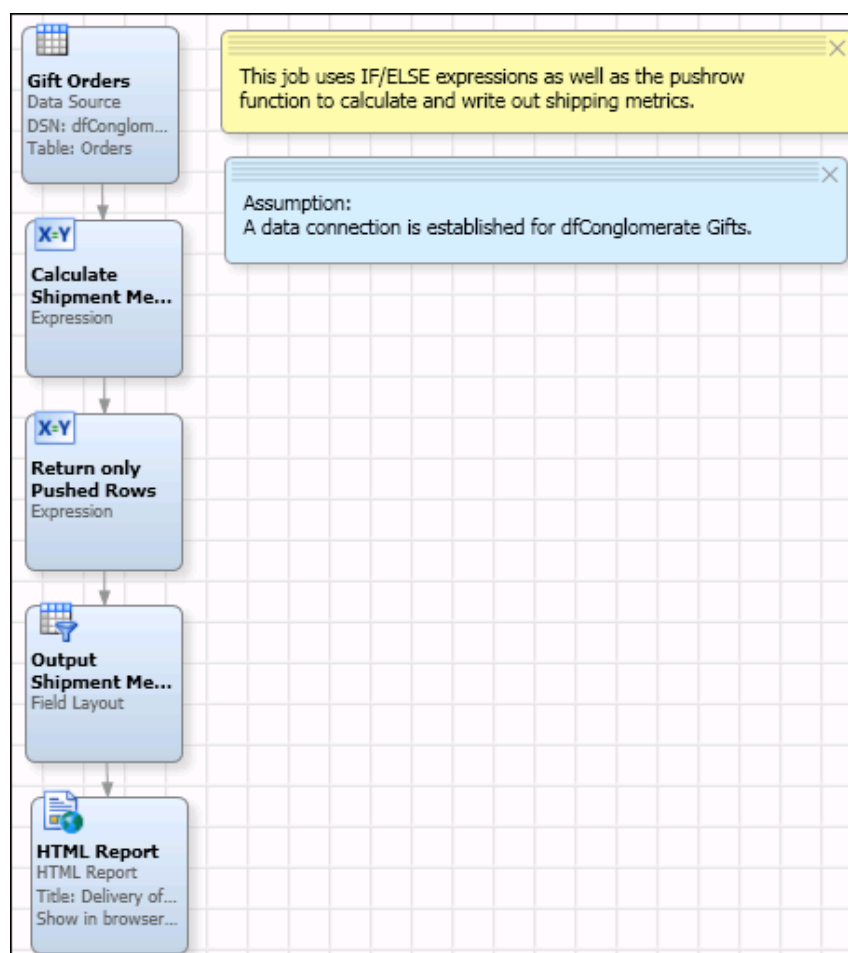


Using the PUSHROW Function

This demonstration illustrates accessing and reviewing a data job that uses IF/ELSE processing in an Expression node to calculate shipping metrics. The pre-processing expression code and the expression code are the same as the previous demonstration. For post-processing, the PUSHROW function is used to push the last record (with the summary statistics) to the stack for output. The **Pushed status field** option is investigated. An additional Expression node is used to write only the last record (the pushed record).

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 **Ch10D3_Pushrow_Example** data job.

The data flow should resemble the following:



3. Review the properties of the Data Source node.
 - a. Right-click the node that is labeled **Gift Orders** and select **Properties**.
 - b. Verify that the selected table is **Orders** from the **dfConglomerate Gifts** data connection.
 - c. Verify that all fields are selected.
 - d. Click **Cancel** to close the Data Source Properties window.
4. Preview the Data Source node.
 - a. If necessary, select **View** ⇒ **Show Details Pane** to toggle **on** the Details pane.
 - b. Right-click the **Data Source** node in the data flow diagram and select **Preview**.
 - c. Verify that there are only 48 records in the data source.
5. Review the properties of the first Expression node.
 - a. Right-click the node that is labeled **Calculate Shipment Metrics** and select **Properties**.
 - b. If necessary, click the **Expression** tab.
 - c. Review the code in the Pre-processing Expression section.

Note: This is the same pre-processing code that is used in the Expression node for the Ch10D1_IF_ELSE_Example data job. The code is shown without the comments.

```
string(10) Shipment_Metric
hidden real count
hidden real late_count
hidden real early_count
hidden real ontime_count
count = 0
late_count = 0
early_count = 0
ontime_count = 0
```

- d. Review the code in the Expression section.

Note: This is the same code that is used in the Expression node for the Ch10D1_IF_ELSE_Example data job. The code is shown without the comments.

```
count = count + 1
if((`SHIPPED DATE` - `ORDER DATE`) > 4) OR
  (isnull(`SHIPPED DATE`) AND ((today() - `ORDER DATE`) > 4))
begin
  Shipment_Metric = 'LATE'
  late_count = late_count + 1
end
else
  if `SHIPPED DATE` == `ORDER DATE`
  begin
    Shipment_Metric = 'EARLY'
    early_count = early_count + 1
  end
  else
    begin
      Shipment_Metric = 'ONTIME'
      ontime_count = ontime_count + 1
    end
  end
```

- e. Review the code in the Post-processing Expression section.

```
// create variables for percentage calculations
real percent_late
real percent_early
real percent_ontime

// calculate percentages and round to 2 decimal places
percent_late = round(((late_count/count) * 100),2)
percent_early = round(((early_count/count) * 100),2)
percent_ontime = round(((ontime_count/count) * 100),2)

// use the pushrow function to push row with calculated values
// on the output stack
pushrow()
```

- f. Click the **Settings** tab.
- g. Verify that **Pushed status field** is selected and the value **Pushed** is specified.

Output

Generate rows when no parent is specified

Pushed status field:

Return status field:

- h. Click the main item on the thread (labeled **Ch10D3_Pushrow_Example**) to return to the Data Flow tab.

6. Preview the Expression node that is named **Calculate Shipment Metrics**.
 - a. If necessary, select **View** ⇒ **Show Details Pane** to toggle on the Details pane.
 - b. Right-click the **Expression** node in the data flow diagram and select **Preview**.
 - c. Verify that there are now 49 records in the preview and that the last record (record 49) seems to be a duplicate of the 48th record.
 - d. Scroll to the right and locate the **Pushed** field.
 - e. Verify that all records except the last one show that **Pushed** equals *False*.
 - f. Verify that the last record shows that **Pushed** equals *True*.
 - g. Also verify that the last record has the calculated values for **Percent_late**, **Percent_early**, and **Percent_ontime**.

	TE	TAX STATUS	Shipment_Metric	Percent_late	Percent_early	Percent_ontime	Pushed
44	0	(null)	ONTIME	(null)	(null)	(null)	False
45	0	(null)	ONTIME	(null)	(null)	(null)	False
46	0	(null)	ONTIME	(null)	(null)	(null)	False
47	0	(null)	LATE	(null)	(null)	(null)	False
48	0	(null)	LATE	(null)	(null)	(null)	False
49	0	(null)	LATE	14.58	18.75	66.67	True

7. Review the properties of the second Expression node.
 - a. Right-click the **Return only Pushed Rows** node and select **Properties**.
 - b. If necessary, click the **Expression** tab.
 - c. Verify that no code appears in the Pre-processing Expression section.
 - d. Verify that no code appears in the Post-processing Expression section.
 - e. Review the code in the Expression section.

```

if Pushed
  return true
else return false

```

- f. Click the main item on the thread (labeled **Ch10D3_Pushrow_Example**) to return to the Data Flow tab.

8. Preview the Expression node named **Return only Pushed Rows**.
 - a. If necessary, select **View** ⇒ **Show Details Pane** to toggle on the Details pane.
 - b. Right-click the **Expression** node in the data flow diagram and select **Preview**.
 - c. Verify that a single row is previewed.
 - d. Scroll to the right and verify that the single row is for **Pushed** equals *True*.

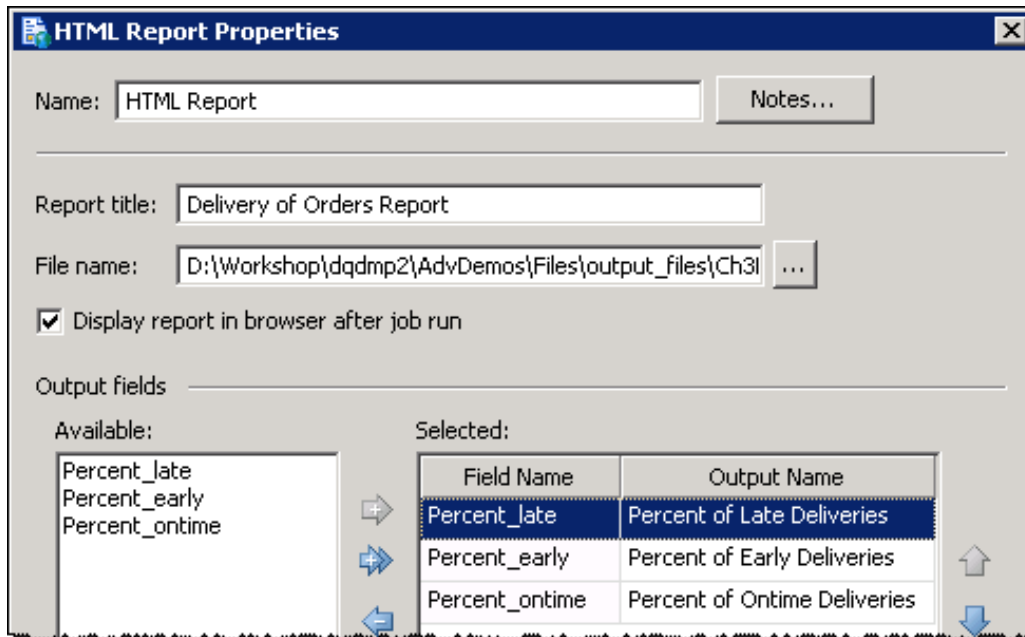
		TAX STATUS	Shipment_Metric	Percent_late	Percent_early	Percent_ontime	Pushed
1	0	(null)	LATE	14.58	18.75	66.67	True

9. Review the properties of the Field Layout node.
 - a. Right-click the **Output Shipment Metrics Calculation Fields** node and select **Properties**.
 - b. Verify that only three fields (**Percent_late**, **Percent_early**, **Percent_ontime**) are in the Selected fields list.

Selected fields:	
Field Name	Output Name
Percent_late	Percent_late
Percent_early	Percent_early
Percent_ontime	Percent_ontime

- c. Click **Cancel** to close the Field Layout Properties window.

10. Review the properties of the HTML Report node.
 - a. Right-click the **HTML Report** node and select **Properties**.
 - b. Verify that a report title and a file name are specified in their respective fields.
 - c. Verify that **Display report in browser after job run** is selected.
 - d. Verify that only three fields (**Percent_late**, **Percent_early**, **Percent_ontime**) are in the Selected list.



11. Run the job and review the output.
 - a. Select **File** ⇒ **Save**.
 - b. Select **Actions** ⇒ **Run**.

The HTML report appears in a browser:

The screenshot shows an Internet Explorer browser window displaying the 'Delivery of Orders Report'. The report shows three columns: Percent Of Late Deliveries (14.58), Percent Of Early Deliveries (18.75), and Percent Of Ontime Deliveries (66.67).

Percent Of Late Deliveries	Percent Of Early Deliveries	Percent Of Ontime Deliveries
14.58	18.75	66.67

- c. When you are finished viewing the report, close the browser.
- d. Select **File** ⇒ **Close** to close the data job.

End of Demonstration

Setup for the Question

Suppose a single record of data is passed in to an Expression node in a data job.

Late	Early	OnTime
7	9	32

For the Expression node, in the Pre-processing Expression section, these two fields are declared:

```
string value
integer count
```

For the Expression node in the Expression section, the following code is specified:

```
value='Late'
count=late
pushrow()
value='Early'
count=early
pushrow()
value='On-Time'
count=ontime
```

114

Copyright © SAS Institute Inc. All rights reserved.



x.01 Short Answer Question

Knowing that the PUSHROW function creates a record or row in the output, how many records are in the result set of the Expression node?

10

Copyright © SAS Institute Inc. All rights reserved.



Solutions to Activities and Questions

x.01 Short Answer Question – Correct Answer

Knowing that the PUSHROW function creates a record or row in the output, how many records are in the result set of the Expression node?

There should be three records in the result set for the Expression node.

- Two explicit PUSHROW functions are executed.
- An implicit PUSHROW action occurs.

Result set of Expression node:

Late	Early	Ontime	Value	Count
7	9	32	Late	7
7	9	32	Early	9
7	9	32	On-Time	32