

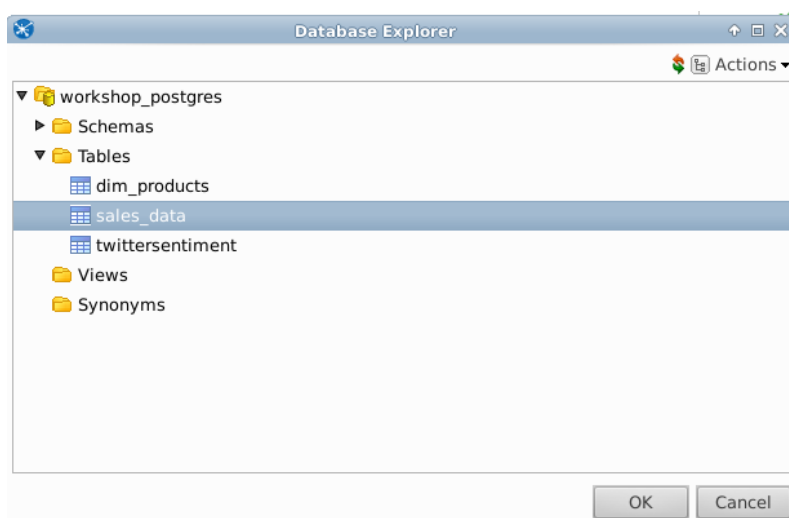
Passing URL Parameters to Kettle Transformations

PDI Remote Invocation

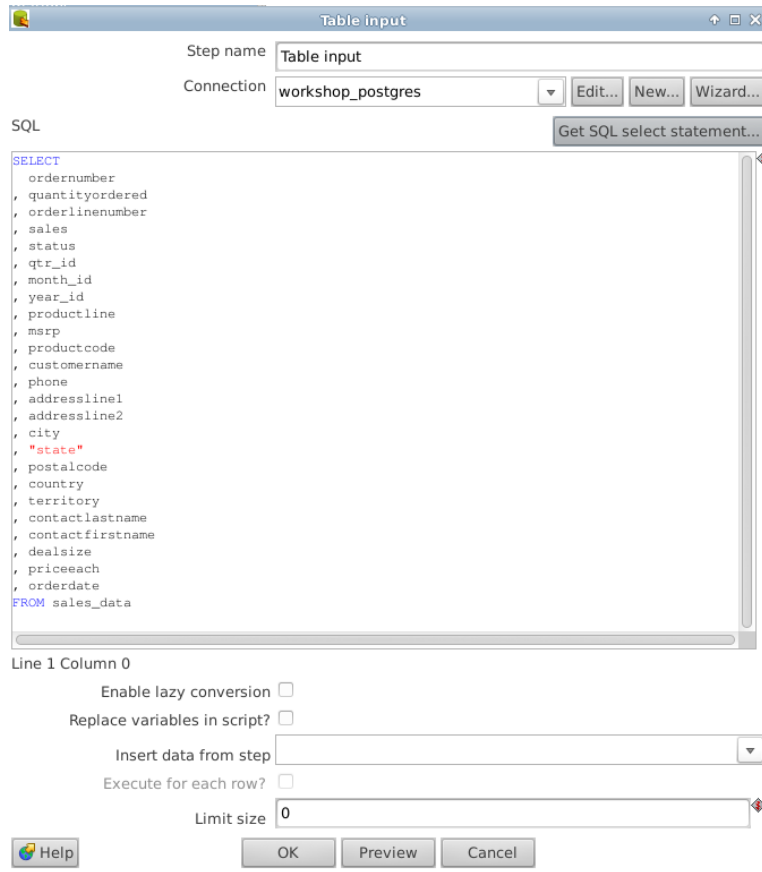
During this exercise we will demonstrate how to pass parameters to kettle jobs and run them via URLs. This is useful for accessing kettle jobs from other applications on a dynamic basis when scheduling does not give enough granularity.

Build the Transformation

1. Open a terminal window and enter: `cd ~/pentaho/`
 - a. Press enter
2. Enter the command: `./ctlscript.sh restart`
 - a. Press enter
3. This will restart DI server in case it is not currently running.
4. Go to Tools -> Repository -> Connect.
5. Select Workshop and enter "password" as the password. Press ok.
6. Open the `pdi_remote_invocation_final` located in the `public/PDI Workshop` for reference purposes.
7. Create a new transformation by selecting File -> New -> Transformation
8. Save your transform as `pdi_remote_invocation` by clicking File -> Save As. Save it into the admin folder.
9. Drag a Table input step onto the canvas from the input folder.
10. Select "workshop_postgres" as the data base Connection and press the Get SQL select statement button.
11. Navigate down to the `sales_data` table and press OK. Press Yes to include column names.



12. The table input step should look like this. Press OK.



The 'Table Input' dialog box is shown. It has a title bar with standard window controls. Below the title bar, there is a 'Step name' field containing 'Table input' and a 'Connection' dropdown menu set to 'workshop_postgres'. To the right of the connection dropdown are buttons for 'Edit...', 'New...', and 'Wizard...'. Below these is a large text area for the SQL query, which contains a SELECT statement listing various columns from a table named 'sales_data'. To the right of the SQL text area is a button labeled 'Get SQL select statement...'. Below the SQL text area, there is a section for 'Line 1 Column 0' with several options: 'Enable lazy conversion' (unchecked), 'Replace variables in script?' (unchecked), 'Insert data from step' (a dropdown menu), 'Execute for each row?' (unchecked), and 'Limit size' (a text field containing '0'). At the bottom of the dialog are buttons for 'Help', 'OK', 'Preview', and 'Cancel'.

Step name: Table input

Connection: workshop_postgres

SQL:

```
SELECT
  ordernumber
, quantityordered
, orderlinenumber
, sales
, status
, qtr_id
, month_id
, year_id
, productline
, marp
, productcode
, customername
, phone
, addressline1
, addressline2
, city
, "state"
, postalcode
, country
, territory
, contactlastname
, contactfirstname
, dealsize
, priceeach
, orderdate
FROM sales_data
```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

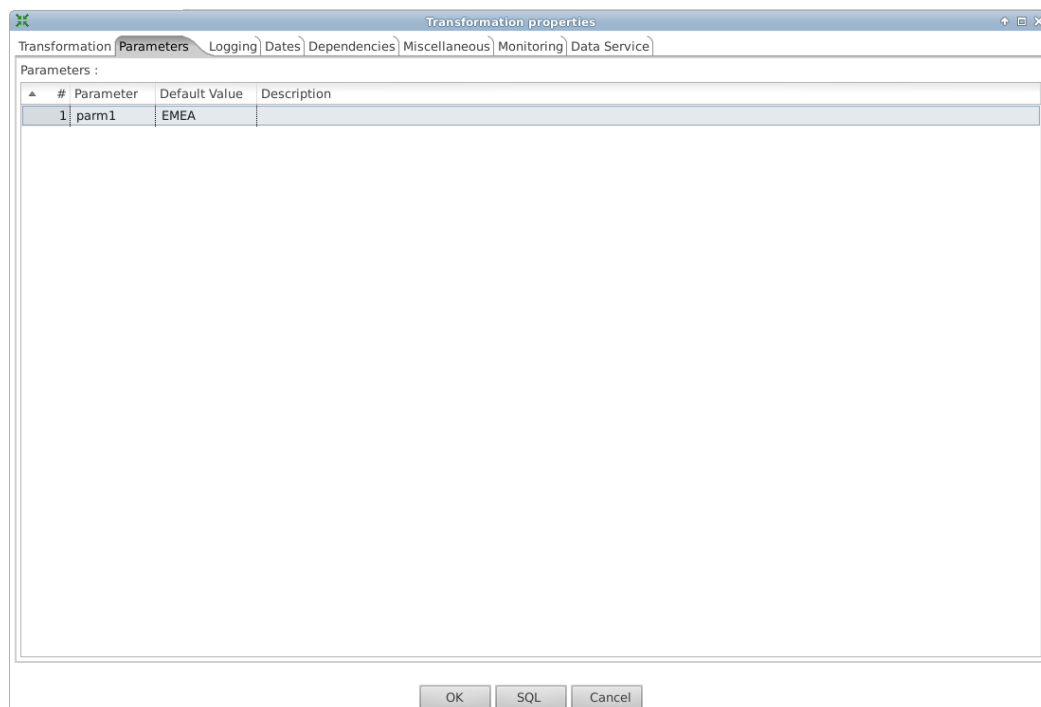
Insert data from step: [dropdown]

Execute for each row? ☐

Limit size: 0

Buttons: Help, OK, Preview, Cancel

13. Right click anywhere on the canvas and select Transformation Properties.



The 'Transformation properties' dialog box is shown. It has a title bar with standard window controls. Below the title bar, there are several tabs: 'Transformation', 'Parameters', 'Logging', 'Dates', 'Dependencies', 'Miscellaneous', 'Monitoring', and 'Data Service'. The 'Parameters' tab is selected. Below the tabs, there is a table with the following columns: '#', 'Parameter', 'Default Value', and 'Description'. The table contains one row with the following values: '#', 'parm1', 'EMEA', and an empty 'Description' cell. At the bottom of the dialog are buttons for 'OK', 'SQL', and 'Cancel'.

Transformation properties

Transformation Parameters Logging Dates Dependencies Miscellaneous Monitoring Data Service

Parameters:

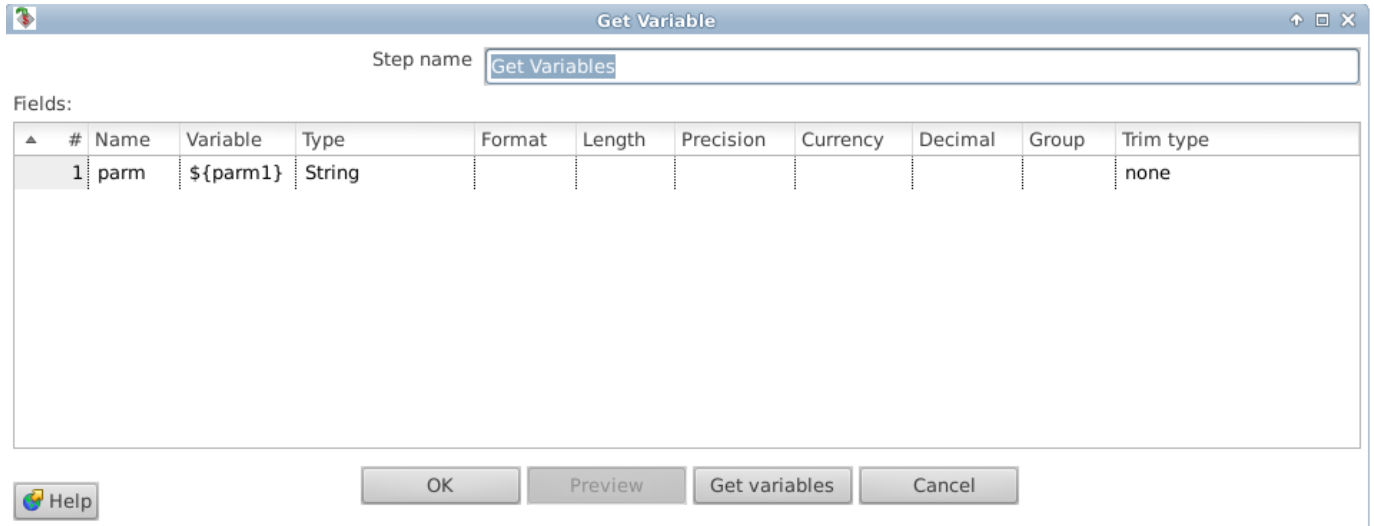
#	Parameter	Default Value	Description
1	parm1	EMEA	

Buttons: OK, SQL, Cancel

14. On the Parameters tab, enter “parm1” with the Default Value of “EMEA”.

Filter by GET Parameter

1. Drag the Get variables step under the Jobs folder on the canvas and attach it to the Table input step. Enter the following information.



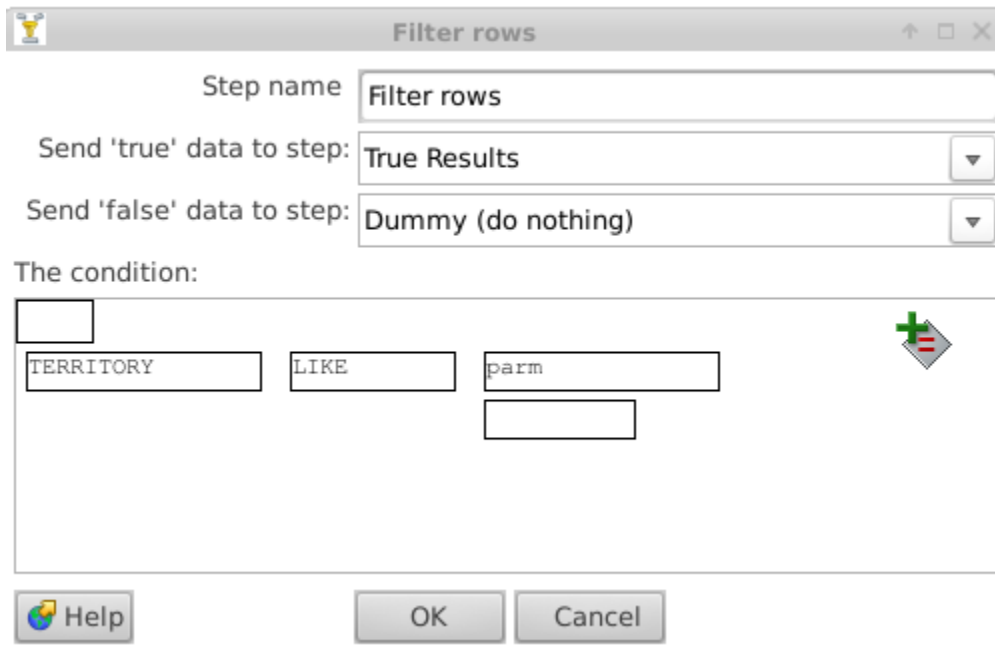
Step name:

Fields:

#	Name	Variable	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	parm	\${parm1}	String							none

Buttons:

2. Drag a Filter rows step from the Flow folder onto the canvas and attach it to the Get variables step. Add the following information:



Step name:

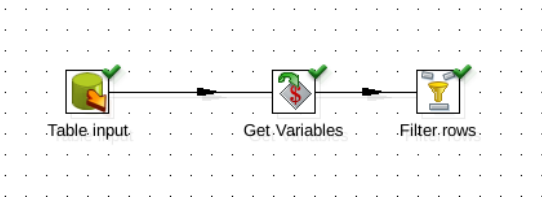
Send 'true' data to step:

Send 'false' data to step:

The condition:

Buttons:

3. At this point the transformation should look something like this:



4. Next, add 2 Dummy steps to the canvas, one below the Filter rows step and one to the right of it.
5. Right click on the bottom dummy step and rename it to “No Match”. Right click on the other dummy step and rename it to “Matching Results”.
6. Connect the Filter step to No Match when Result is false and connect it to Matching Results when Result is True.
7. Add a JSON output step to the canvas and connect it to the Matching Results dummy step.
8. Update it as follows, making sure to check the servlet checkbox.

Step name:

General Fields

Operation:

Settings

Json bloc name:

Nr rows in a bloc:

Output Value:

Compatibility mode: ☐

Output File

Filename:

Append: ☐

Create Parent folder: ☐

Do not open create at start: ☐

Extension:

Encoding:

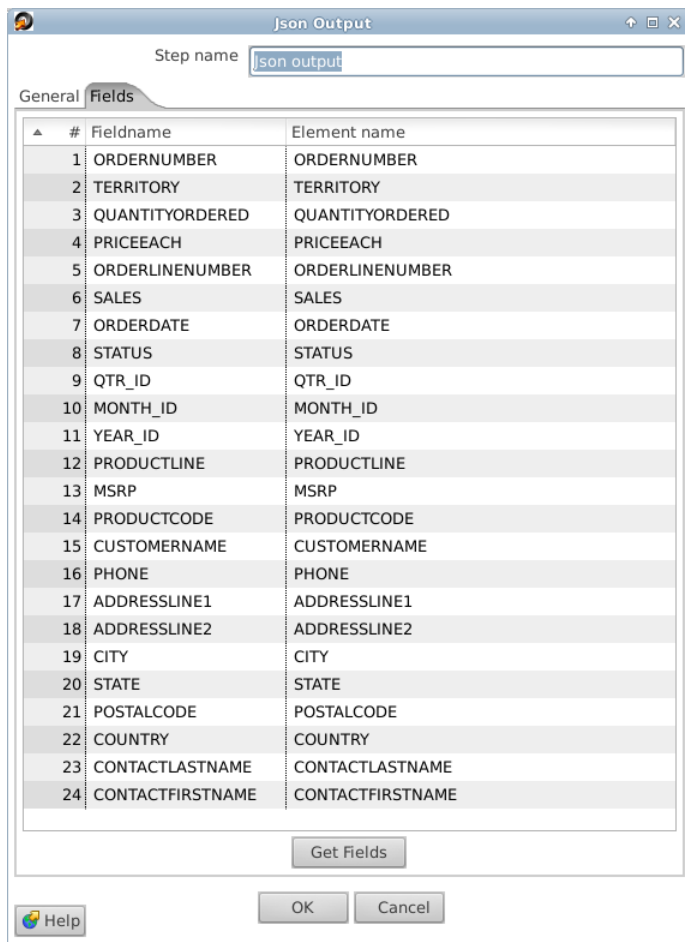
Pass output to servlet: ☒

Include date in filename?: ☐

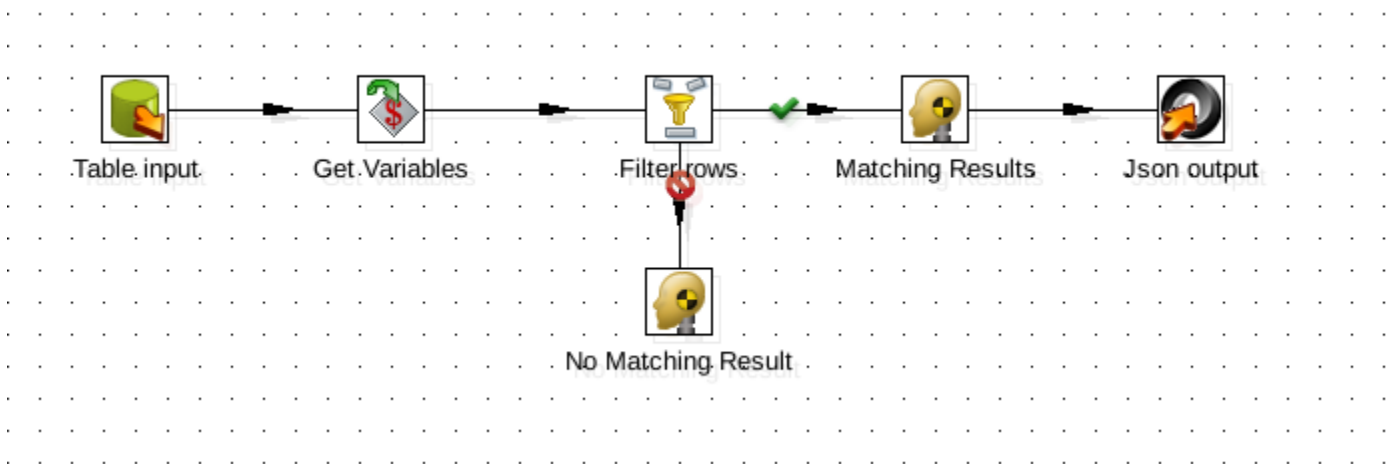
Include time in filename?: ☐

Add File to result filenames: ☐

9. Click on the Fields tab of the JSON output step and press Get Fields.



10. The transformation should look like this. Save it and run it to preview data.



11. Navigate to the following URL: http://localhost:9080/pentaho-di/kettle/executeTrans/?rep=Workshop&trans=/public/PDI%20Workshop/pdi_remote_invocation_final&parm1=APAC

Enter admin/password if prompted for user name and password.

12. You will see JSON output for EMEA data. By changing parm1=NA in the URL, the data will update to show only North America Territory sales data.

Review Exercise #5

What We Covered...

- Created parameterized transformation for running from URL
 - JSON Output
 - Filter on Territory