

# KNIME™ Analytics Platform Getting Started Guide

With the Actian DataFlow™ Free Node Pack and  
Parallel DataFlow Executor for KNIME

## About this Guide

### If you are completely unfamiliar with KNIME

This Getting Started Guide will help you learn the KNIME interface, learn how to create and edit workflows, and how to use and configure the pre-built operators called nodes. In general, it will give you a good introduction on using the KNIME open source data mining platform to do data preparation and analytics workflow design, testing and execution.

### If you are already familiar and comfortable with KNIME

This guide will familiarize you with the Actian DataFlow free node pack, and the DataFlow executor. It will teach you the difference between flowable and non-flowable nodes, when and how to use each, how to use the DataFlow executor, and how to use both the DataFlow executor and KNIME default executor in a single workflow.

### What is DataFlow?

The free DataFlow executor and Actian DataFlow nodes are available for download on the KNIME website. They are intended for use with large data sets or compute intensive analytics jobs that tend to overload or bog down hardware resources. They are optimized to take best advantage of modern hardware's parallel processing capabilities.

### When Would I Want to Use DataFlow?

The DataFlow nodes and executor can speed up workflow execution times on virtually any hardware platform, from a laptop to a Hadoop cluster. If you have workflows with execution speed issues, this information can be very useful. It is not uncommon to shorten workflow execution times from hours or days to minutes on the same hardware by switching to flowable nodes and the DataFlow executor.

DataFlow also gives you the option of designing Hadoop workflows in the KNIME user interface, rather than having to write MapReduce or other code to define Hadoop workflows. DataFlow generally executes between 10 and 100X faster than MapReduce on the same hardware and is fully integrated with YARN.

### Requirements:

- A Windows, Mac or Linux computer with at least 4 GB of RAM, 8 GB preferred, and at least 4 GB of available hard drive space.
- On Mac, 64 bit operating system is required. On Windows and Linux, 64 bit operating system is preferred.
- The KNIME application, version 2.9.x or 2.10.x, with the Actian DataFlow addition, 6.5.x. (DataFlow is not yet compatible with version 2.11 of KNIME.)
- The DF Executor Get Started workflow zip file.
- The DF Get Started sample data zip file.
- This set of instructions.

## Setup

This chapter gives you all the information needed to set up your system and be ready to design analytic workflows with KNIME and DataFlow. After completing these steps, you will have a functioning KNIME workbench, workspace, standard KNIME nodes, DataFlow nodes and both the default and DataFlow executor.

- General Information about KNIME and Actian DataFlow
- Steps to set up KNIME
- Steps to add DataFlow functionality

### Components

- KNIME workbench and a base set of nodes
- Actian DataFlow nodes and desktop DataFlow executor for KNIME
- Sample data for tutorial

## General Information

### KNIME

KNIME is an open source analytics platform with a user-friendly graphical workbench that can be used for creating ETL or data preparation workflows, data analysis workflows, and machine learning workflows.

KNIME is freely available for download at <http://www.knime.org/knime>

Commercial versions of KNIME software with additional functionality are also available at <http://www.knime.org/>

### KNIME Nodes

The Eclipse-based KNIME workbench lets you build fully functional working applications by stringing together and configuring individual pre-built operators (nodes) with data flow directional arrows (edges). KNIME and the KNIME partner network have created over 1000 nodes with a vast array of functionality.

KNIME is completely platform agnostic, providing versions for Mac, Linux and Windows. DataFlow nodes are one set of partner created nodes that provide a subset of functionality in highly parallel form, making them particularly useful with very large data sets and parallel cluster computing execution environments.

Nodes can be used to accomplish a huge variety of functions including:

- Extracting data from files and databases
- Loading data into files and databases
- Executing SQL statements
- Statistical analysis – Mean, Median, SD ...
- Deriving new values
- Transforming data and data types
- Machine learning – training, testing and using
- Visualization
- And a whole lot more

## Action DataFlow

In partnership with KNIME, Actian has contributed three functionalities to the free KNIME download. (Starting in May, 2015)

1. DataFlow Node Pack – A set of data preparation, manipulation, evaluation, analysis and machine learning operators that are all optimized to run highly parallel for highest possible execution speed on modern hardware platforms.
2. DataFlow Executor – A parallel execution engine that can be used in place of the default KNIME execution engine in order to execute KNIME workflows in a pipelined and parallel optimized fashion.
3. Flowable API – A set of operations that can alter non-DataFlow KNIME nodes to be executable with the DataFlow Executor so that they can take advantage of the execution speed boost. This API has been a part of the KNIME node development standards for a few years now. New KNIME nodes should be created with the flowable capabilities from this point forward. A skilled Java programmer can alter an existing KNIME node generally in just a few hours to make it flowable.

The DataFlow executor provides scalability to KNIME by using all the server cores and cluster computers that are available at runtime. With Actian DataFlow, a workflow created on a development server, or a laptop computer, can be deployed for execution onto a large production server or even a cluster without any design changes. The DataFlow operators and any KNIME operators that have been made flowable will automatically use the extra resources, such as cores and memory, that are available in the production environment.

## Limitations on the Free Version of DataFlow

The free version of Actian DataFlow that is available on the KNIME website is not limited in terms of time or functionality. The only limit is on the number of parallel streams the executor will handle. The free desktop version is limited to 2 streams of parallelism and the free cluster version is limited to 10 streams. Each stream can spawn multiple parallel execution threads.

The free desktop version of DataFlow will maximize the data processing capabilities of most laptop or desktop computers. The free cluster version will maximize the data processing capabilities of most clusters with 4 compute nodes or less, assuming about 4 cores per node. If more parallel data processing power is needed, contact [sales@actian.com](mailto:sales@actian.com) or [info@actian.com](mailto:info@actian.com) for information regarding the commercial version.

## Steps to Set Up KNIME

### Steps to Add DataFlow Functionality

Set up Preferences?

## Sample Data and Workflows

### General Information

The first workflow you will create contains two different kinds of operators, which are called nodes in KNIME. The Delimited Text Reader is an Actian DataFlow node and the Interactive Table is a flowable node that was not created by Actian. Since both of these nodes are flowable, this workflow can be executed by both the default KNIME and DataFlow executors. There are advantages to both executors in different situations. Once we have added more nodes to the workflow, we will look more closely at how the two executors work.

For now, the main thing to remember is:



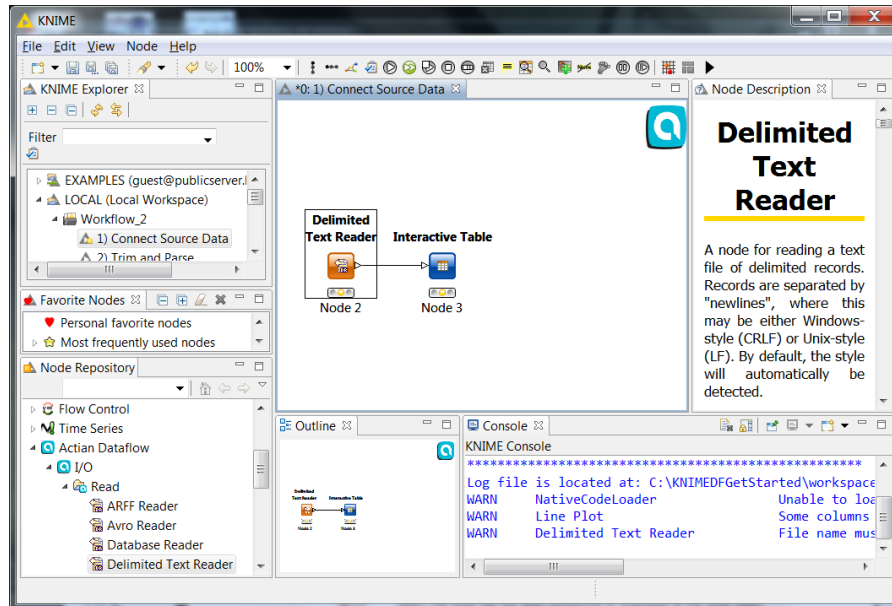
**Only Actian DataFlow nodes and other flowable nodes can be executed by the DataFlow executor.**

The majority of existing nodes in KNIME are not flowable at this time. With the exception of the Interactive Table, which we know is flowable, we will stick to Actian DataFlow nodes for most of this lesson. When we get to a part of the workflow that requires functionality not available in DataFlow nodes, then we'll look at how the two types of nodes, flowable and non-flowable, and the two executors, KNIME default and DataFlow, can be mixed to accomplish your goals.

### Step 1: Create a New Workflow Project

In KNIME, everything you do is contained in a workflow. Before we can do any work, we need a new workflow project.

1. Start the KNIME.exe.
2. On the File menu, select New.
3. Select Actian DataFlow Workflow, and then click Next.
4. Change the default workflow name to whatever you like. "Workflow\_1," for instance. Leave the Workflow Group set to the default value. Leave the profile set as the default, Development.
5. Click Finish. A new tab titled something like "Workflow\_1" will appear. Also, your workflow will appear on the left side of the interface in the KNIME Explorer.



**Frequently Asked Question:** What's the difference between a KNIME workflow and a DataFlow workflow?

**Answer:** The main difference is that a DataFlow workflow uses the DataFlow executor unless you specify otherwise. A KNIME workflow uses the Default executor unless you specify otherwise.



**(Optional) Bonus Steps:** Switch workflow from DataFlow to Default and back.

1. Right click on the workflow name in the KNIME Explorer, and choose Configure.
2. Select the Job Manager Selection tab. Where it says "Job Manager for this node," click the down arrow and choose <<Default>>.
3. Click OK. Now, it's a KNIME workflow.
4. Do steps 1 and 2 again, but choose the DataFlow job executor, and it's back to being a DataFlow workflow.



**Tip:** Look in the upper right corner of your canvas. If you see an Action style "A" then your workflow is set to use the DataFlow executor. If not, it's set to use the Default executor.



**WARNING:** Do NOT change the executor on individual nodes! This only causes problems and is almost never useful. You will see the one exception to that rule toward the end of this tutorial.

## Step 2: Add a Text Reader Node

The first step in nearly any analytics workflow is to read in some data. Our sample data is in a fairly common pipe delimited format.

1. In the Node Repository located at the lower left of the KNIME workbench, expand the Action section to get the DataFlow nodes. Expand the I/O section and Read section.
2. Select the Delimited Text Reader node and drag it onto the workflow canvas.
3. Right click on the Delimited Text Reader node on the canvas and select Configure from the context menu. (Or, just double-click on the node.)
4. Click on the Browse button to go find the sample data file.
5. Select the “20140313.txt” file from the sample data directory.
6. This is a pipe delimited file. Choose the pipe | from the Field Separator drop-down list.
7. You can see in the preview window that the first row in this file is not data, but column names. Check the box for Has Header Row. You will see the Preview window change to indicate that the way the reader node will read the data is now different.



**Note:** The first four fields visible in the Preview window are not in the data file. They are all information available to the system that is added for you automatically by the Reader.

8. All of our files for these exercises will be local. Check the box “Read file locally instead of in cluster.”
9. Click OK to save your settings and close the configuration window.



**Note:** You’ll see that a yellow indicator is now under this node where before it was red and had a little yellow triangle sign. Red indicates a node that cannot be executed, either because it hasn’t been configured, or because there is an error. Yellow means ready.

**Delimited  
Text Reader**



**Delimited  
Text Reader**



**Note:** When you click on the Delimited Text Reader, the documentation for that node appears in the upper right corner of the KNIME workbench interface. All KNIME nodes have this feature. This is super handy for figuring out what unfamiliar nodes do, and how to use them.



**Note:** If you were connecting to a Hadoop data set, you would not indicate a file, just a directory. Or, you might use a wild card to indicate only a sub-set of files. If KNIME was not on the cluster itself, and you needed to indicate a Hadoop file set that was not local, that would require a few other extra steps. See “Integrating with Hadoop” in the docs for details: <http://help.pervasive.com/display/DF632/Integrating+with+Hadoop>



#### **Bonus steps: Explore the raw data**

To get the information about your file that you need to know in order to configure a Delimited Text Reader node, you can explore the file, or a small sample of the file if it is very large, in any text editor.

1. On your computer, navigate to KNIMEDFGetStarted\data and open the file SourceDataSample10Rows with a text editor such as NotePad ++ or TextPad or VI.
2. Observe the delimiters, "" around the data fields and | defining the borders between fields.
3. Also, notice that the top row is field names.

### Step 3: Add an Interactive Table

Interactive Table is a useful node that allows you to look at your data, as well as sort and search it. Interactive Table is not a DataFlow node, but it is a flowable node, and therefore it can be executed with the DataFlow executor. We will use this node a lot to check on our progress as we go along.

1. In the Search box at the top of the Node Repository, type in Interactive. You will see the nodes below shift until only nodes with that word in them are visible.
2. Click on the Interactive Table node and drag it onto the canvas.
3. Click and hold on the tiny triangle on the right side of the Delimited Text Reader and drag to the triangle on the left of the Interactive Table. Release and an arrow should appear connecting the two nodes.



**Tip:** If you have selected a node on the canvas and you double-click a new node in the Node Repository, it will jump onto the canvas and connect itself to the selected node.





**Note:** Unlike the default KNIME executor, the DataFlow executor executes in a pipeline which does not allow checking on data in the middle of a workflow. The Interactive Table node can be very useful for checking to make certain that data manipulation is happening as expected.

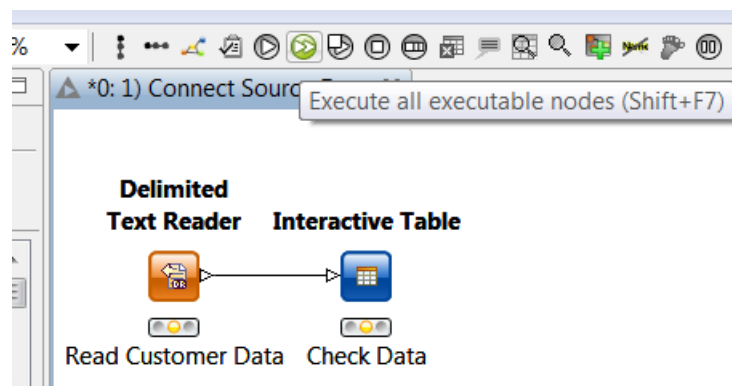



**Bonus Steps:** Name the nodes.

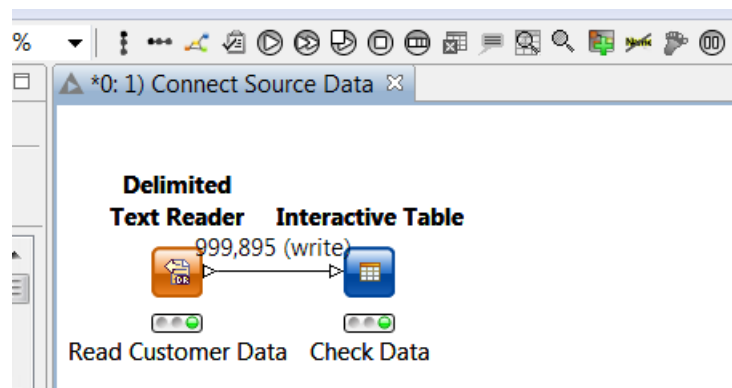
1. Click on the box under the Delimited Text Reader node that says “Node 1.”
2. Click again to get a cursor to appear in the box. (Double clicking also works.)
3. Type in a name for that node, such as “Read Customer Data.”
4. Click on the box under the Interactive Table node. Click again.
5. Type in a name for that node, such as “Check data.”

## Step 4: Execute the Workflow

The workflow should have both nodes indicating yellow, ready to execute, now.



1. Save the workflow by selecting Save from the File menu or clicking the Save icon. 
2. On the toolbar, click the >> in a green circle “Execute all” button.
3. The execution begins.
4. If it completes without error, the indicators at the bottom of each node will turn green. Green means executed.





**Note:** While executing, a DataFlow workflow will show a count of the records along the arrow. Multiple nodes along the flow will execute simultaneously. Arrows will show dotted between nodes that are still executing and turn solid when done. In a KNIME workflow, one node executes at a time unless the flow branches, and the indicators under the nodes will show percentage complete and other information.



**Tip:** If there is an error, the indicator on the node with the problem will turn red. This can be very useful when there are many nodes in a workflow and you are trying to pinpoint a problem. The error itself will appear in the KNIME Console at the bottom of the workbench.



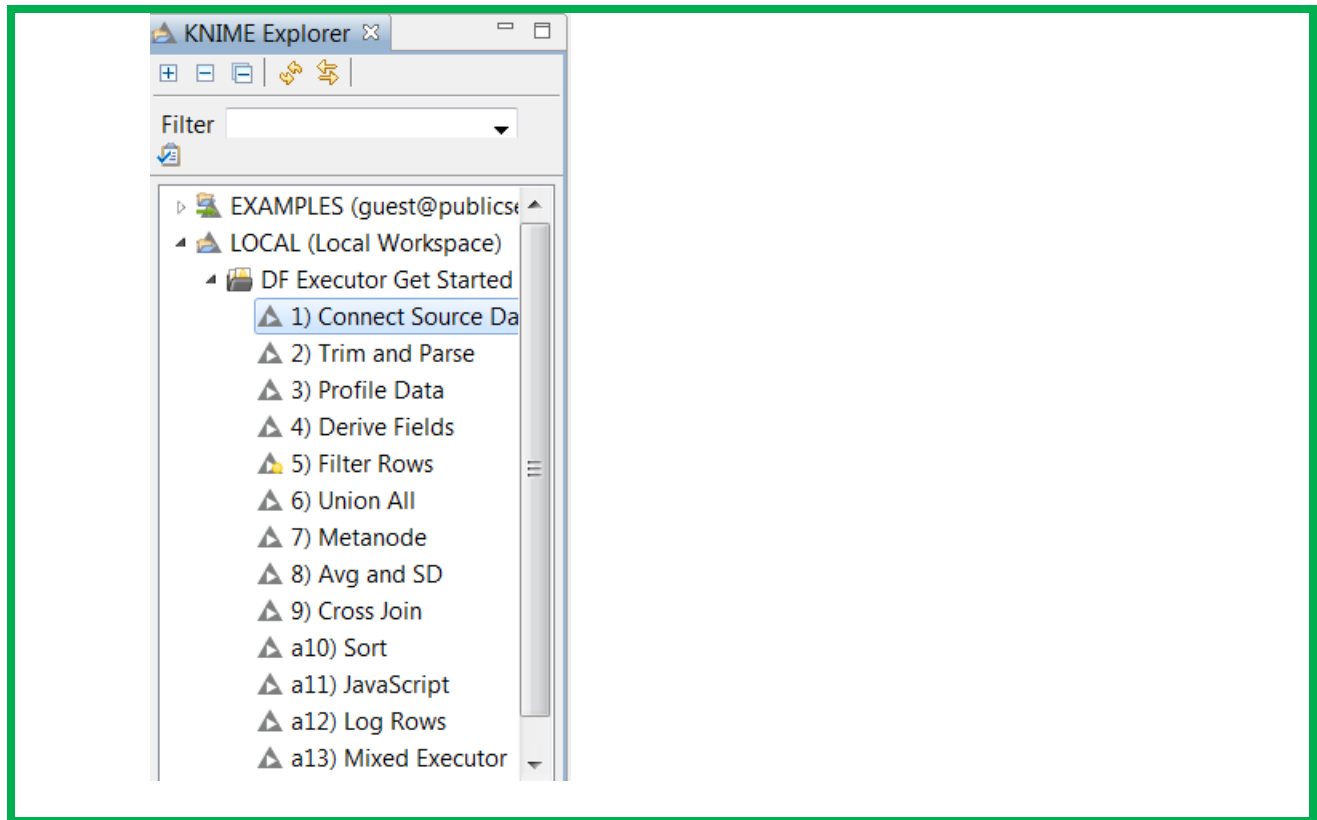
**Bonus steps:** Reset and rerun a workflow

1. Right click on the Delimited Text Reader and choose Reset from the menu. This will set all indicators back to yellow, and make the workflow ready to run again.
2. Click Execute All again.
3. As we add more nodes, experiment with resetting workflows at different nodes.
4. You can also reset a workflow by right clicking on the workflow name in the KNIME Explorer and choosing Reset.



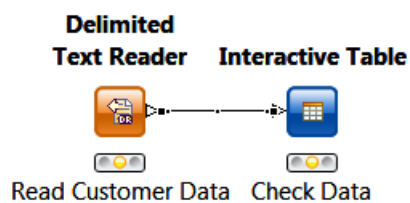
#### **SAVE POINT: Connect Source Data**

Save points are points in the tutorial where the workflow has been saved independently in the solution set. If you get stuck, you can jump forward or back to the nearest save point and continue working from there. Below is a list of the Save Points in this tutorial.

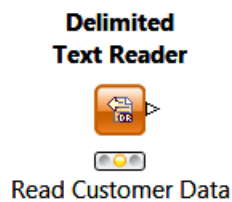


#### Step 4: Get the workflow ready to add more nodes

1. Select the arrow that connects the Delimited Text Reader and the Interactive table, and click Delete on your keyboard to remove it.



2. Click the Interactive Table node and delete it for now. We'll add it back in later.



## Step 5: Add the Trim Whitespace node

The Trim Whitespace node removes leading and trailing spaces from data. There are several ways to do this in KNIME. The Trim Whitespace node is the most convenient way to do it for many fields at once.

Trimming white space can solve a lot of problems before you encounter them, so it is often a good practice. Many text comparisons or other operations on data will fail or get erroneous results if there are spaces at the beginning or end of the data. For example:

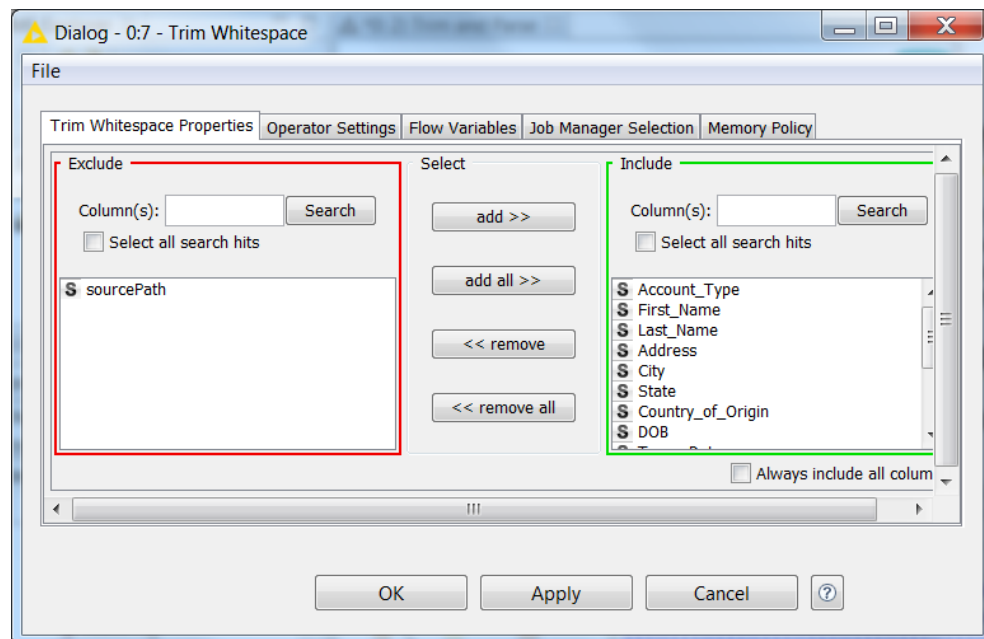
City == "San Antonio" will fail to match if the data is actually " San Antonio "

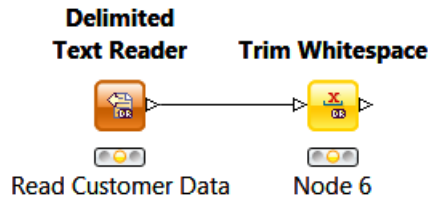
Also, data type conversions may fail if leading or trailing spaces are included in non-text data. For example:

cDate(" 20150409 ") will fail with an invalid date error.

cDate("20150409") will succeed.

1. In the Node Repository, expand the Action DataFlow nodes. (You can also search for "Trim" but be sure to choose the DataFlow "Trim Whitespace" node. Non-DataFlow nodes won't work with the new executor.)
2. Expand Transformations and Manipulation.
3. Drag the Trim Whitespace node onto the canvas.
4. Connect the Delimited Text Reader node to the Trim Whitespace node.
5. Right click on the Trim Whitespace node, and choose "Configure."
6. Select the "sourcePath" field name.
7. Click Remove. There's no need to trim that one. It's set by the system and will never have leading or trailing spaces.
8. Click OK to save settings and close the window.





**Frequently Asked Question:** Will the Trim Whitespace node eliminate spaces inside my data?

**Answer:** No. If you use this node on a City field, for example:

“ San Antonio ” → Trim Whitespace → “San Antonio”

Not “SanAntonio”

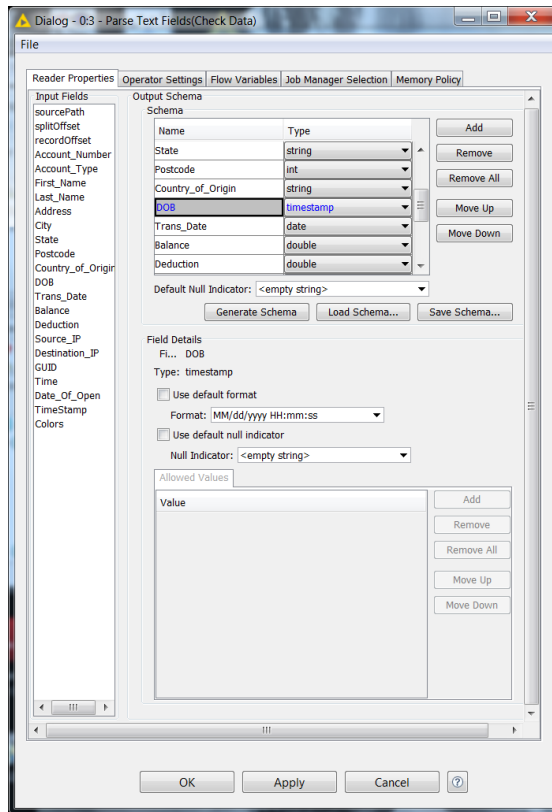
## Step 6: Add the Parse Text Fields Node

The Delimited Text Reader takes its best guess at field data types and other metadata information. Sometimes, its best guess is wrong, and sometimes, you need to adjust the way the data is parsed to match the needs of your workflow. The next node will allow us to do that. Here is the schema that we wish to define:

Name	Type
Account_Number	int
Account_Type	string
First_Name	string
Last_Name	string
Address	string
City	string
State	string
Postcode	int
Country_of_Origin	string
DOB	timestamp
Trans_Date	date
Balance	double
Deduction	double
Source_IP	string
Destination_IP	string
GUID	string
Time	time of day
Date_Of_Open	timestamp
TimeStamp	timestamp
Colors	string

1. Under Action DataFlow and I/O, select the Parse Text Fields node and drag it onto the canvas.

2. Connect the right triangle on Trim Whitespace to the left triangle on Parse Text Fields.
3. Right click and choose Configure. (Or double click.)
4. Select a field, such as DOB. Click the down arrow and choose the correct data type.
5. In the case of DOB, the correct data type is timestamp.
6. Select the next field to change. (Notice the default and null indicator options as you go along. We're leaving them all to default settings, but you could put a default null value here, change the mask on an unusual date time field, or several other options.)
7. When all fields are set to the correct data types, Click OK to save and close the window.



**Note:** It is possible to change data types and other metadata by clicking the Edit Schema button on the configuration window of the Delimited Text Reader. Changing data types there can save a step later, but it has limitations that the Parse Text Fields node does not. For example, Parse Text Fields will simply split data into data it can parse using the rules defined and data it can't. The Delimited Text Reader may error and fail if some data cannot be parsed using the defined rules.



**Note:** On string data types, there is a trimmed or untrimmed setting. If we had not done the Trim Whitespace node already, we could trim our text fields here. However, for the numeric or

date/time fields, not having the fields already trimmed might cause some of our data type conversions to fail.



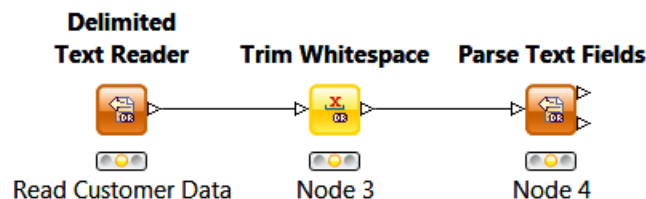
**Frequently Asked Question:** Why does the Parse Text Fields node have two output ports?

**Answer:** If there is a problem parsing the incoming records, records that parsed successfully will pass through the upper port, and records that did not parse successfully will be rejected and pass through the lower port. For example, if one of the incoming DOB fields contains data that is not a valid date, and can't be put into a timestamp data type, the record containing that DOB value would pass through the bottom port.



**Bonus Steps:**

1. Connect an Interactive table to the top Parse Text Fields output port.
2. Connect an Interactive table to the bottom Parse Text Fields output port.
3. Execute the workflow.
4. Right click on the Interactive Table and choose View: Table View on each one to see how many records were not parsed, and see if you can figure out why specific records could not be parsed.



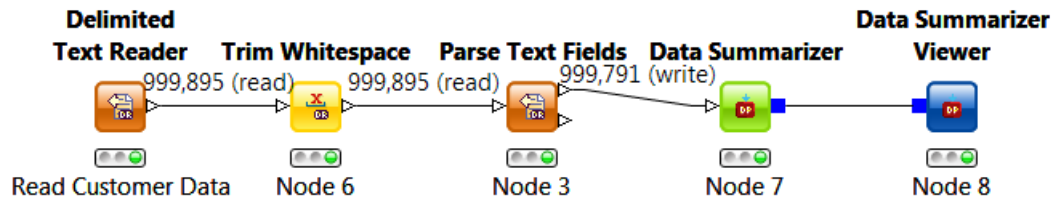
**Save Point: Trim and Parse**

## Step 7: Profile Data

An important step in any analytics workflow is getting a quick, exploratory view of the data. The Data Summarizer provides basic statistics on a data set. The Data Summarizer Viewer displays those in graphic form.

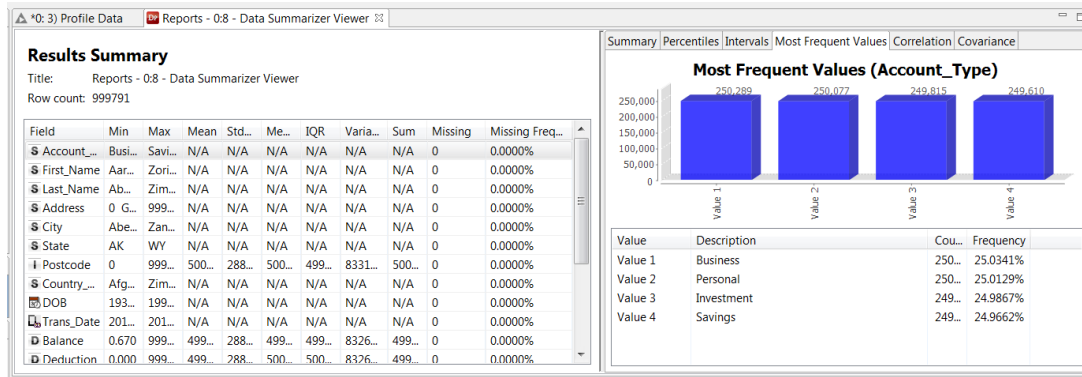
1. Under Actian DataFlow, Data Explorer, choose the Data Summarizer and drag it onto the canvas.
2. Be sure to connect it to the upper port, not the lower, on the Parse Text Fields Operator.
3. Right click and choose Configure.

- By default, all fields will be selected for statistical summaries. The Account Number and GUID fields are not sensible fields for statistical analysis. Select them and click Remove. Click OK.
- Drag the Data Summarizer Viewer onto the canvas.
- Connect the blue square port on the Data Summarizer to the blue square port on the Data Summarizer Viewer. No configuration is necessary for the Viewer.
- Save the workflow and execute it.



**Note:** About 100 records were rejected. You can see the difference in the record count before and after the Parse Text Fields node.

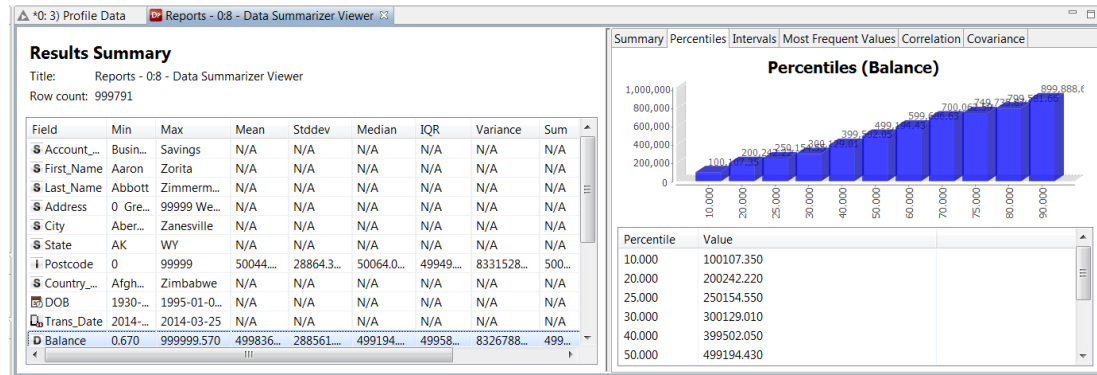
- Right click on the Data Summarizer Viewer and choose View: Reports. This will open a new tab with information about your data.
- Click on the new tab and explore a bit.



**Note:** Account\_Type is a text field with limited values. Most of the summary information is N/A or blank because it only applies to numeric fields. The main thing you can tell is that all the data is there. There are 0 records with missing data in this field. Also, you can see the distribution of the values is very uniform. You can tell this at a glance by looking at the histogram on the Most Frequent Values tab.

- Click on the Balance field to see a good example of summary data for a numeric field.





**Teaser question:** (Teaser questions give you a chance to stop and think. The answer isn't supplied.) What is unusual about this data? What would jump out at you from this quick view of the data if you were working on an analysis project that used it?



**Frequently Asked Question:** What's the difference between a blue box port and a white triangle port on a node?

**Answer:** The blue port is a PMML port, not a data port. KNIME nodes can have many ports, and many different kinds of ports. White triangle ports are always ports that pass data into or out of KNIME nodes. Blue square ports pass statistical models stored in the form of PMML. Many KNIME nodes can read or write PMML, making it possible to pass models from node to node and even from KNIME to other applications that support PMML, such as SAS, Zementis, R, Weka, etc. and from those applications back to KNIME.



**Bonus Steps:** Get summary statistics on the rejected records

1. Drag another Data Summarizer and Data Summarizer Viewer node onto the canvas.
2. Connect them up to the bottom, reject data port on the Parse Text Fields node.
3. Remove the Account Number and GUID fields in the Data Summarizer configuration window.
4. Right click on the Delimited Text Reader, the first node in the workflow, to reset the workflow.
5. Execute the workflow again.
6. Right click on the Data Summarizer Viewer and choose View: Reports.



**Teaser question:** Based on the information you now see, why were those approximately 100 records rejected?

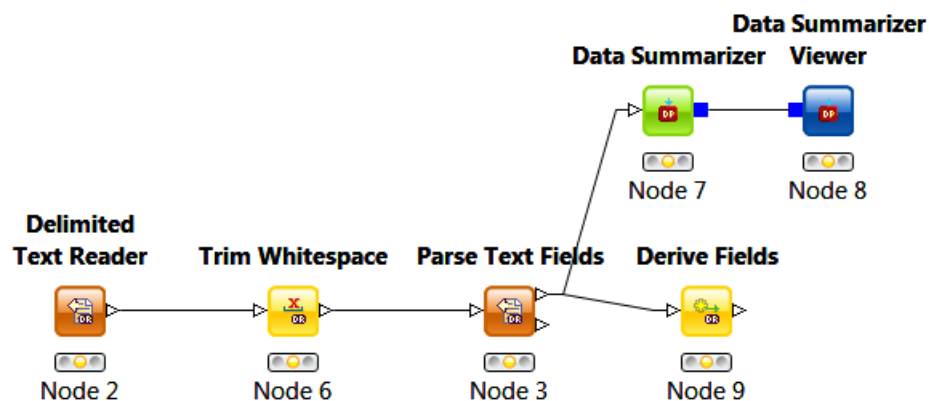


### Save Point: Profile Data

## Step 8: Balance Minus Deduction

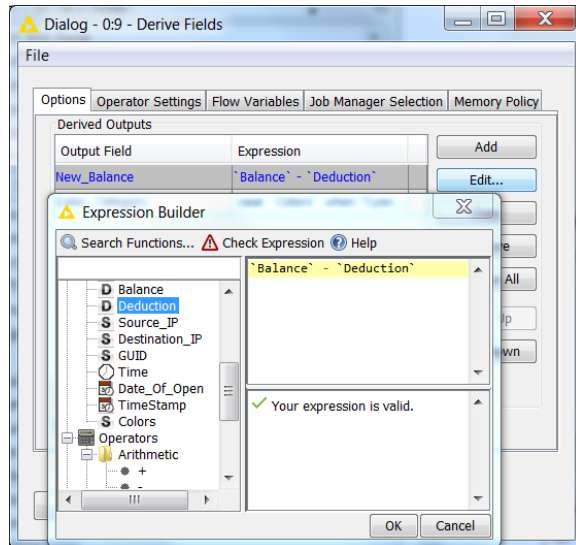
Often, for data analysis, it is useful to mathematically derive a new column from existing data. For example, when working on a churn prevention project, the price of monthly phone base charge and the actual monthly charges is highly correlated, and therefore the second variable is not useful in predicting if the customer will churn. However, the difference between those two charges might be highly predictive.

1. Under Action DataFlow, Transformations, Manipulation, choose Derive Fields. Drag it onto the canvas.
2. Connect the Derive Fields to the top port of the Parse Text Fields node. Yes, two nodes can both be connected to a single output port.



3. On the configuration window for the Derive Fields node, click Add. This will create a new field with the default name of Field0.
4. Double click in that location and replace Field0 with New\_Balance.
5. Click Edit.
6. This will open the Expression Builder window. This is a handy helper window for writing DataFlow script expressions.
7. Select and delete the default field value of 0.
8. On the left column, expand out the Field References.
9. Double click on the Balance field. This will place the field name delimited by back ticks into the Expression. (Those are back ticks, not apostrophes. Back ticks are generally found on the same key as the ~ on a standard keyboard, just above the Tab key.
10. Expand the Operators list, then the Arithmetic operators list. Double click on the – for subtraction.
11. Go back to the Field References list and double click on Deduction. You should now have the expression: `Balance` - `Deduction`

12. Click Check Expression in the upper part of the window. If all is well, a green check mark should appear at the bottom of the window next to the words “Your expression is valid.”
13. Click Ok to close the Expression builder, but leave the Derive Fields configuration window open.



**Frequently Asked Question:** Can't I just type in the expression?

**Answer:** Yes. The reference on the left is really just to help you get comfortable with writing DataFlow script expressions, and help prevent syntax errors from irritating you. The DataFlow script is very similar to JavaScript. If you are familiar with JavaScript, you will pick it up extremely easily.



**Frequently Asked Question:** Then why don't I just use JavaScript, since there's a Run JavaScript DataFlow node?

**Answer:** The DataFlow script expression language is optimized to execute in a highly parallel environment. On the surface, the syntax is designed to look and feel like JavaScript in order to make it easy and familiar to use. The actual execution code is very different. The main difference is that it is highly performant, far more performant than JavaScript. It is, however, a very limited language, with only a small set of functions. If you need functionality you can't get from any node, and can't get from the DataFlow script, then feel free to use the Run JavaScript node. Just be cautioned that it will not perform as well as DataFlow script, even when executed with the DataFlow executor.



**FAQ:** Can I use any other language in KNIME with the DataFlow executor, besides DataFlow script and JavaScript?

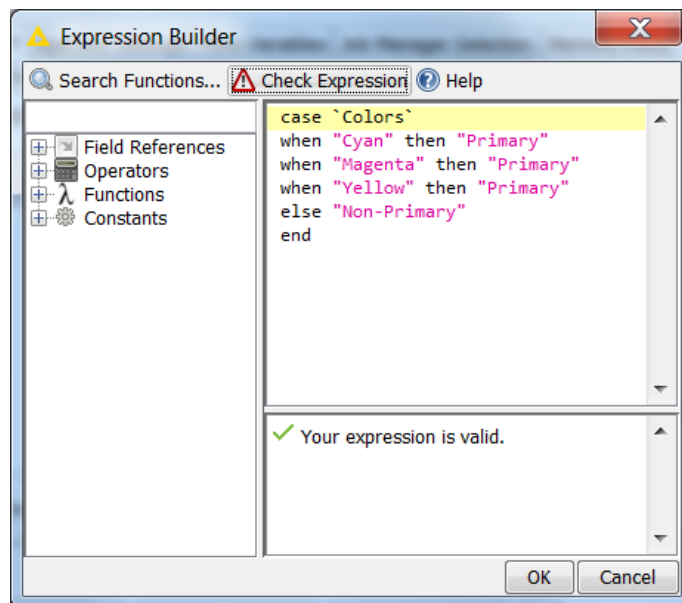
**Answer:** Yes. There is a Run R Snippet Dataflow node and a Run Script node which will execute Python code. Be warned that R is not parallel. It is recommended to do the majority of your data preparation work using DataFlow nodes and only use R if you have a custom algorithm, or other specialized bit of work to do that there are no pre-built nodes for. Most KNIME nodes execute faster than R, even non-DataFlow nodes. Put just the small amount of special R that you need in the Run R Snippet node. The same holds for Python, although it does tend to execute faster than R.



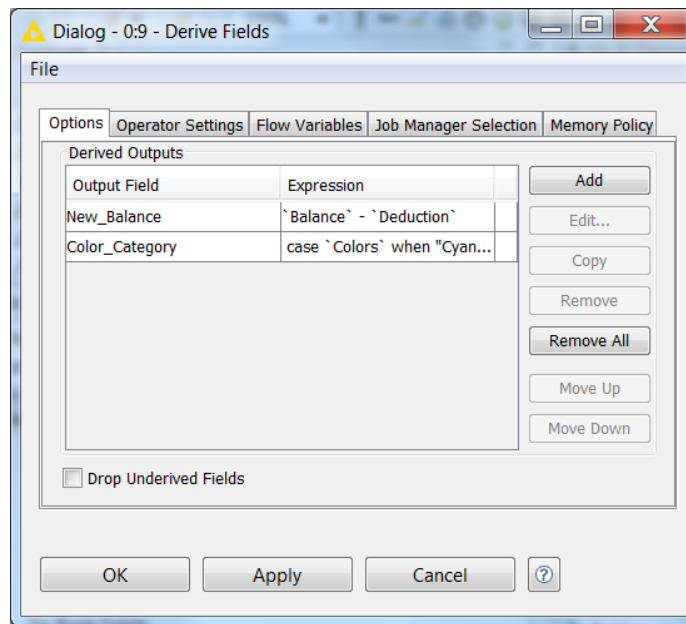
**Tip:** If you need to run JavaScript, use the Run JavaScript node, not the more general Run Script node. The Run JavaScript node is better optimized for executing JavaScript and will give you better performance.

## Step 9: Color Category

1. Click Add in the Derive Fields configuration window to add another new field.
2. Change the field name to Color\_Category. Click Edit.
3. Delete the 0 and type in this expression:



4. Check to make sure your expression is valid. (Did you use back ticks on the `Colors` field name? Do you have straight double-quotes around all your strings? Did you remember to put "end" at the end?)
5. When it's valid, click OK. And Click OK again. We're done deriving fields.



6. Put an Interactive Table node on the canvas and connect it to the Derive Fields output port. (Type “Interactive” in the search field to find it.)
7. Save and execute the workflow.
8. Right click on the Interactive Table, select View: Table View, and note the new fields appearing in your data.



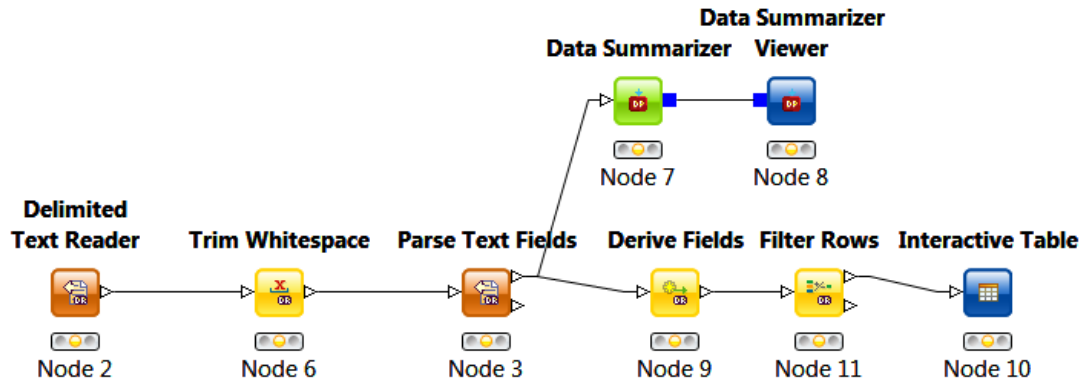
**Save Point: Derive Fields**

## Step 10: Filter Rows

Certain records in your data may not contain information that you are interested in for this particular analysis, or, they may be unsuited to the analysis in some way. A good example would be if you were trying to detect cybersecurity intrusions. Any network traffic that went from one IP to the same IP would be uninteresting, since it could not possibly be an intrusion attempt. In this step, we'll filter out all rows that have source and target IPs the same.

1. Disconnect the Interactive Table node and slide it over to make room.
2. Under Transformations, Filter, drag a Filter Rows node onto the canvas.
3. Connect it to Derive Fields.
4. Click Add in the Derive Fields configuration window.
5. Choose Source\_IP in the Field drop down list.
6. Choose Not Equals <FIELD> under the Comparison operator.
7. Choose Target\_IP under the Field or Value column.
8. Connect the Interactive Table to the Output port, the top port.

- Save and execute the workflow. Note how many records were rejected for having identical source and target IPs.



**Tip:** If you can't remember which port is the output port and which port is the reject port, just hover your mouse pointer over them, and a pop up will tell you.



**Save Point: Filter Rows**

## Step 11: Read in New Compressed Data File

It is often important to combine multiple data sets when working on an analytics project. We're going to read another data set and do a Union All. There are many different ways to join two data sets, left joins, inner joins, etc. We will look at a cross join in a later step. The Union All node essentially appends new data to the end of existing data. In order to do this, the node must be able to match up the correct columns.

- Delete the arrow between the Delimited Text Reader and the Trim Whitespace node. Move the nodes aside to make some space.
- The Union All operator is under Transformations, Aggregate. Drag it onto the canvas.
- Connect the Delimited Text Reader to the upper input port on the Union All node.
- Connect the Union All output port to the Trim Whitespace node.
- Drag a new Delimited Text Reader node onto the canvas. Click on the default name, Node #, and type in a new name, Read Compressed Data.
- Double-click to configure the node. The new data is in a compressed format. You can find it here: `KNIMEDFGetStarted\data\20140318.txt.bz2`
- This is a pipe | delimited file, so set the delimiter value.



**Note:** In the Preview window, you will see the data. Notice that, while this is similar data, the fields are not in the same order, nor is there a header to indicate field names. You will have to provide a schema so that the data can be read correctly and matched up with similar columns in the other file. We will do that in the next step, so leave the window open.



#### **Bonus Steps: Define and Save a Schema as a File**

1. Click the Edit Schema button. The Configure Schema window will open.
2. Type in the field names to define the schema below:

<b>Name</b>	<b>Type</b>
Account_Number	int
Postcode	int
Country_of_Origin	string
Trans_Date	string
TimeStamp	string
Colors	string
First_Name	string
DOB	string
Date_Of_Open	string
Balance	double
Deduction	double
Account_Type	string
Source_IP	string
Destination_IP	string
GUID	string
Time	time of day
Last_Name	string
Address	string
City	string
State	string

3. Click Save Schema, and navigate to the KNIMEDFGetStarted\schemas directory.
4. Name the file something that makes sense.
5. Use your schema instead of the one provided in Step 12.

## **Step 12: Re-Use a Saved Schema**

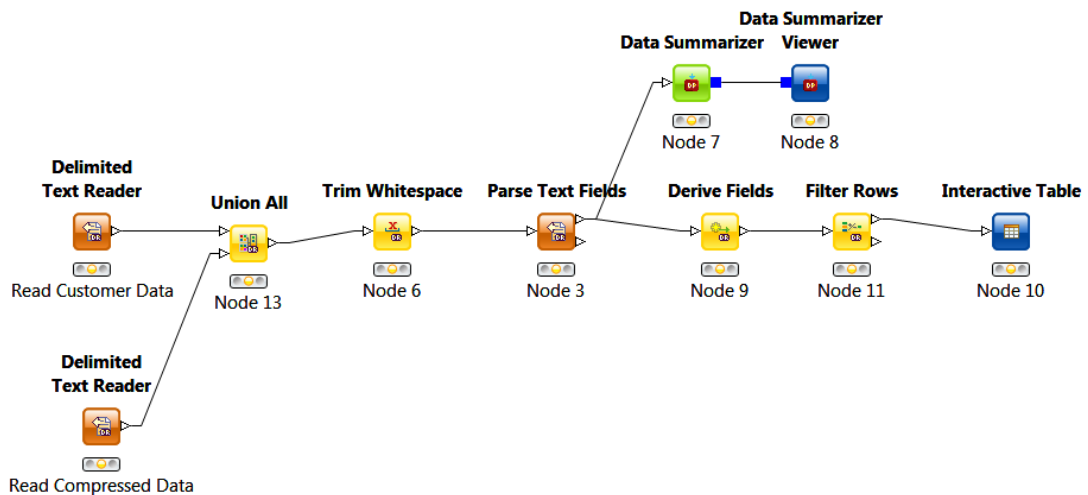
In a real data analysis job, you would have to define the schema, possibly by guess work, possibly by using a definition from somewhere else, as we did in the bonus steps above. That has already been done for you in this case.

1. Click the Edit Schema button. The Configure Schema window will open.
2. Click the Load Schema button.
3. Browse to KNIMEDFGetStarted\schemas\CompressedTextSchema. Click Open.

This will load the schema:

Name	Type
Account_Number	int
Postcode	int
Country_of_Origin	string
Trans_Date	string
TimeStamp	string
Colors	string
First_Name	string
DOB	string
Date_Of_Open	string
Balance	double
Deduction	double
Account_Type	string
Source_IP	string
Destination_IP	string
GUID	string
Time	time of day
Last_Name	string
Address	string
City	string
State	string

4. Click OK to close the Configure Schema window. Notice that the fields are now correctly named in the Preview window. Click OK to save the settings and close the configuration window.
5. Connect the Read Compressed Data node to the bottom, source2, data port on the Union All node.
6. Save and execute the workflow.







### Save Point: Union All



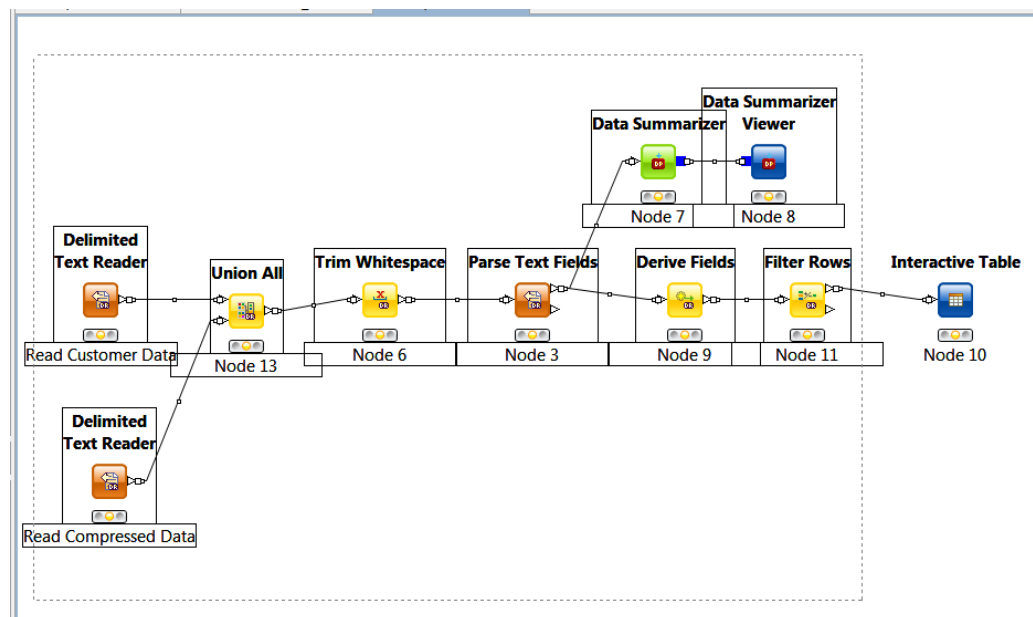
#### Bonus Steps: Look at a Random Sample of Your United Data

1. Under Transformations, Filter, drag a Random Sample node onto the canvas.
2. Connect the Union All output port to the Random Sample input port.
3. Drag an Interactive Table onto the canvas. (Or, you can copy the existing one and paste another onto the canvas.) Connect it to the Random Sample node.
4. Configure the Random Sample node to use Relative Sampling of a .01%. You don't need to set a Sample size unless you set the mode to Absolute, and you can leave the Random Seed at the default setting.
5. Execute the workflow and look at the data in the Interactive Table. The compressed data will have 0 as the split setting. Otherwise, it should be seamlessly appended to the original customer data file.

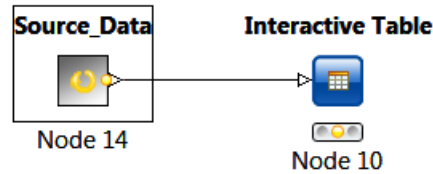
## Step 13: Collapse Multiple Nodes Into One Metanode

A metanode is similar to a sub-process. It collapses multiple workflow steps into a single step in another workflow. This can simplify workflows for explanation purposes, and can also separate DataFlow nodes from nodes that need the default KNIME executor. We will see more of that later. For now, let's create a metanode.

1. Using the mouse, draw a box around all the nodes except the Interactive Table. This will select everything within that area.



2. Right click on one of the selected nodes and choose Collapse Into Meta Node. A window will pop up to give you a chance to enter a name for the metanode.
3. Call it Source Data.
4. Click OK.



5. To expand and look at what's inside the metanode, double click on it. A new tab will open with the contents of the metanode. You can continue to work with it just as before.



**Save Point: Metanode**



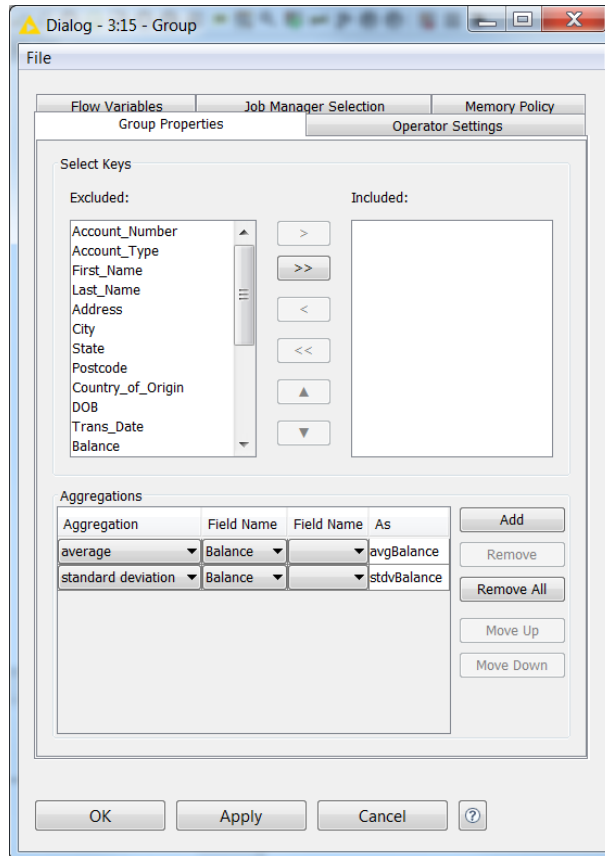
**Bonus Steps:** If you decide you no longer want a metanode, you can expand it back to a normal part of the workflow.

1. Right click on the metanode.
2. Choose Expand Meta Node. This reverts the workflow to the way it was before.
3. Select everything but the Interactive Table.
4. Right click on one of the nodes, and choose Collapse Meta Node.
5. Name it Source Data and click OK.

## Step 14: Average and Standard Deviation

It is often helpful to group multiple records down to a single aggregate value. For example, if analyzing smart meter data, you may have readings for every minute, but an average energy usage per hour would be plenty of granularity to see the usage patterns. We're going to group our data by average balance and standard deviation in balance.

1. Delete the arrow between the Source Data metanode and the Interactive Table, and move Interactive Table aside to make a little room.
2. The Group node is under Transformations, Aggregate. Drag it onto the canvas in between the metanode and the Interactive Table, and connect them.
3. In the configuration window, click Add.
4. Under Aggregation, choose average.
5. Under the first Field Name field, choose Balance. You can leave the second one blank, since we are only aggregating on one field.
6. Under As, put the name of the new field to create. We'll call it avgBalance.
7. Click Add again.
8. Choose standard deviation for the Balance field, and name the new field stdvBalance.



9. Click OK.
10. Save and execute the workflow.
11. Have a look at the Interactive Table to see the output.

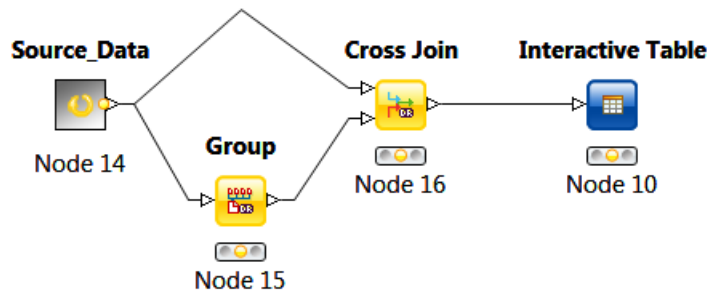


**Save Point: Avg and SD**

## Step 15: Cross Join

To include the grouped data, balance and standard deviation, with the original data, we will need to join the two. A Cross Join will add those two summary fields to every row in the data.

1. Disconnect the Group operator from the Interactive Table.
2. Connect the Source Data metanode to the upper input port on the Cross Join operator.
3. Connect the Group node to the lower input port on the Cross Join operator.
4. Connect Cross Join to the Interactive Table.
5. No configuration is necessary for this node. Save and Execute.



6. View the data. AvgBalance and stdvBalance will be the last 2 columns.



**Note:** The top input port on any Join node is always the left input and the bottom is the right input for determining left and right joins. If you forget which is which, you can always hover the mouse over the port and KNIME will tell you.



**Note:** This guide has shown you how to use all of the Aggregate DataFlow nodes except Join, which is probably the Aggregate node you will use the most often. The Join node lets you do the standard SQL style joins that you would be accustomed to using in a database. Inner, Left Outer, Right Outer, Full Outer and Hash joins can all be done with the Join operator. Key fields can be defined in the configuration window and even predicate expressions, such as where clauses, can be added.

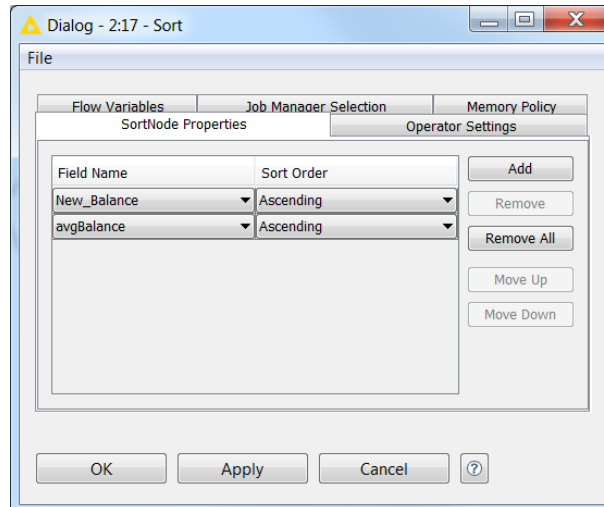


**Save Point: Cross Join**

## Step 16: Sort

Sorting data is a very common and useful thing to do in analytics workflows. However, it pays to keep in mind that it is very compute intensive. In particular, when you have large distributed data sets, sorting can be an action which causes huge performance bottlenecks. Because of this, use Sort sparingly in your workflows and place it with forethought. If you are going to filter out a fair amount of your data, then do that before sorting. Once data is sorted, the data will be preserved in sorted order. This means that later operations in the workflow can assume that the data will be in that order.

1. Under Transformations, Manipulation, you'll find the Sort node. Place it in the workflow between the Cross Join and Interactive Table nodes.
2. Configure the Sort node to sort on two fields, New\_Balance and avgBalance, both in Ascending order.



3. Save and Execute the workflow.
4. View the data.



**Tip:** Some nodes need data in sorted order so that they can perform their tasks. If the data is not sorted, they will automatically sort the data. If you sorted the data in a previous workflow, then use the Assert Sorted node under Transformations, Manipulation. This operator lets downstream nodes know that the data has already been sorted and on what keys. This can vastly improve performance by preventing automatic sorting when it isn't necessary.



**Save Point: Sort**

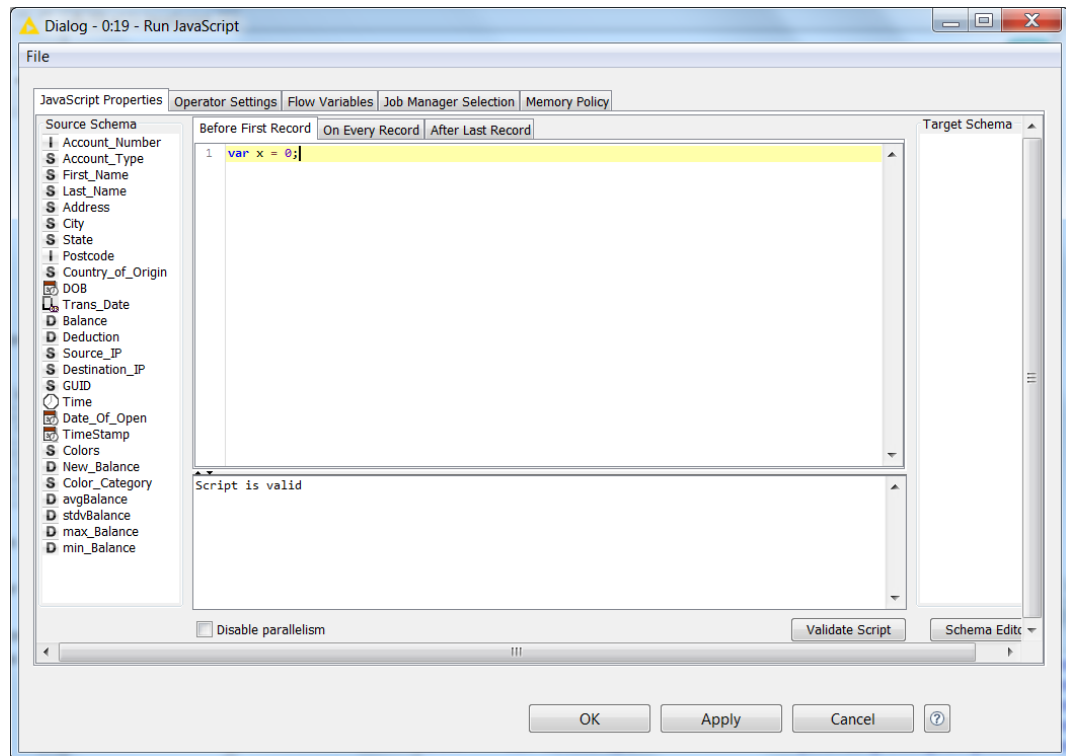
## Step 17: Run JavaScript Code

KNIME and KNIME partners and community, including Actian, have created more than 1000 nodes. This gives KNIME a tremendous breadth of functionality. In spite of that, there will occasionally be nuggets of functionality that you need for a specific workflow that are not available in any pre-built operator. You could, of course, build a KNIME node yourself to do that task. However, if you are not a skilled Java developer, this could be a little daunting.

If you can write the functionality you need in Python, JavaScript or R, you can still have it in a KNIME workflow. Use an Actian DataFlow node to let the DataFlow executor run that code. DataFlow can sometimes divide the work across nodes to help optimize the script code, but in general, these scripts will execute much more slowly than other DataFlow nodes. So, just like Sort nodes, use scripting nodes sparingly and with forethought.

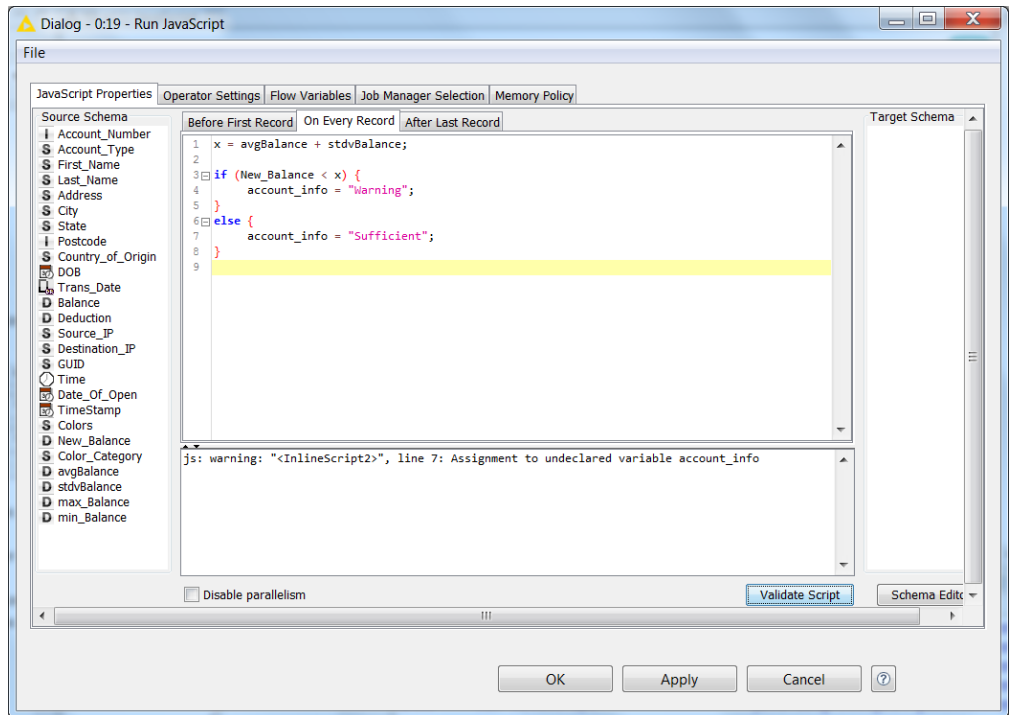
In this example, we would like to add a “Warning” field if the New\_Balance field is less than one standard deviation below the average balance for that account. If not, the flag will be set to Sufficient.

1. Place the Run Javascript node in the workflow between Sort and Interactive Table.
2. In the configuration window, on the Before First Record tab, add:  
`var x = 0;`

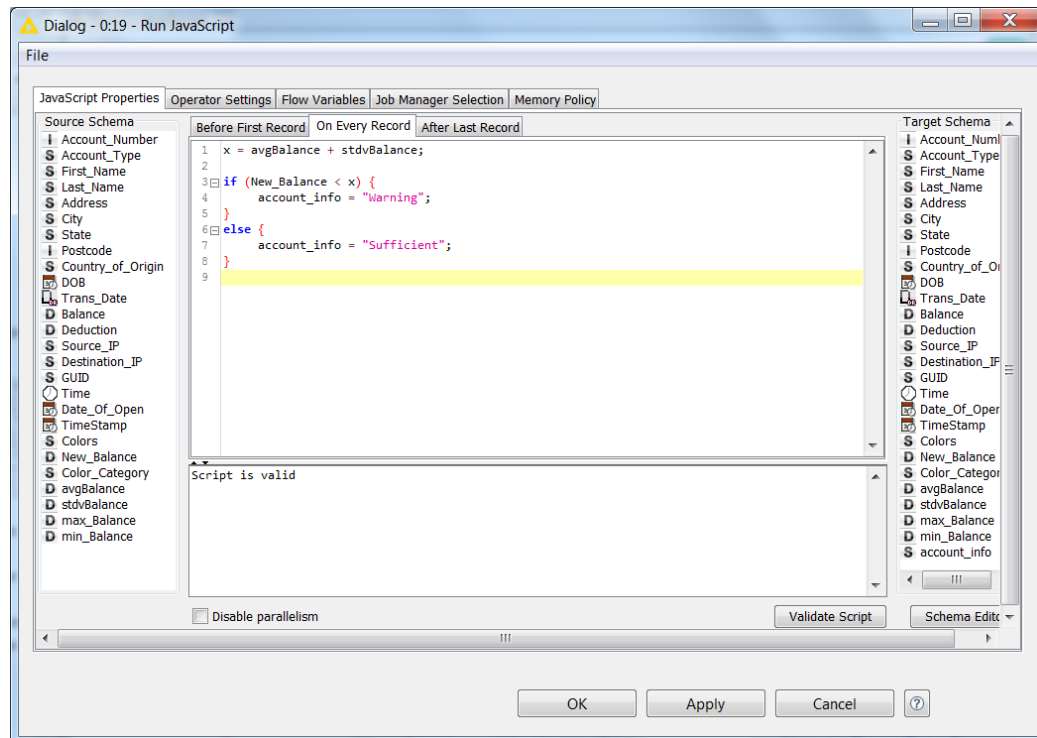


3. Click the Validate Script button. "Script is valid" should appear in the bottom of the window.
4. On the On Every Record tab, type in this code:  
`x = avgBalance + stdvBalance;`

```
if (New_Balance < x) {  
    account_info = "Warning";  
}  
else {  
    account_info = "Sufficient";  
}
```



5. When you click Validate Script, you'll get a warning telling you that account\_info doesn't exist yet in the target data's schema.
6. So, let's add it. Click the Schema Editor.
7. Click the Generate Schema button. This will give you an identical schema to your source data set.
8. Scroll to the end of the field name list and click the Add button.
9. Change the default name to account\_info and click OK on the schema editor. The new list of target fields will appear in the Target Schema list on the right.
10. Click Validate Script. "Script is valid" should now appear.



11. Save and run your workflow. Take a look at your data. The new data field should appear in the last column in the Interactive Table.



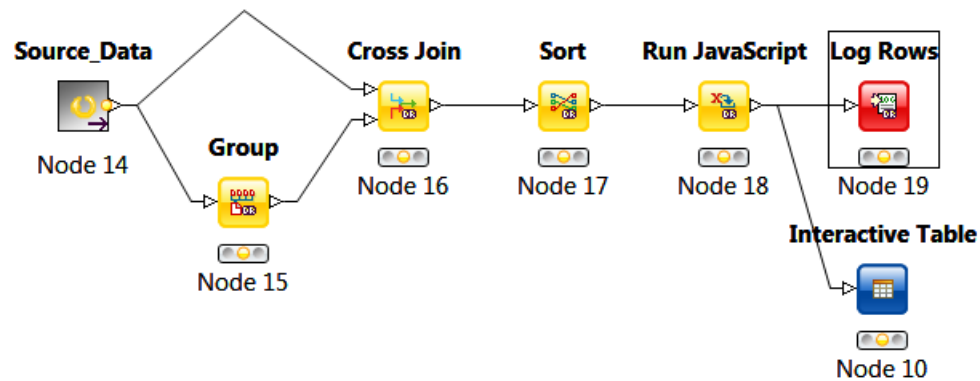
**Save Point: JavaScript**

## Step 18: Log Rows

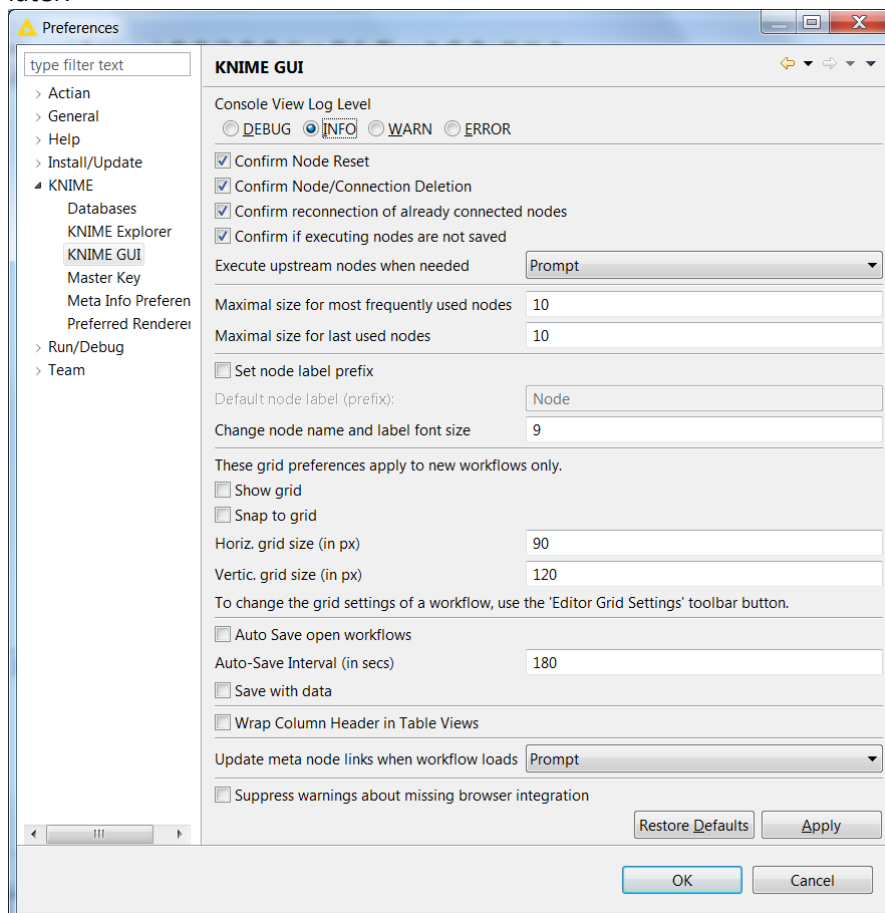
In many cases, when trying to figure out what is going wrong in a large scale project, it is helpful to get a look at what your data is doing. The Log Rows operator, under I/O, lets you log a row of data to the KNIME console every so many records. This way you can monitor the data, and possibly pinpoint where a data problem might exist.

1. Place a Log Rows operator at the end of the workflow. Leave the Interactive Table and connect the output from the JavaScript node to the Log Rows node also.





2. Set Log Frequency to 10,000.
3. A lot of KNIME functionality is in the Preferences window. We'll need to set the logging preferences for the KNIME interface in order to see the rows. On the File menu, choose Preferences.
4. Expand the KNIME preferences and set Logging to Info. This will give you a lot of information in the console. If it becomes too much, you can set it to Warn or Error only later.



5. Save, execute and watch your KNIME console at the bottom of your screen while it is executing. Double click on the top of the console tab to fill your screen with the console. Double click the top of the tab again to return to the previous view.



Save Point: Log Rows

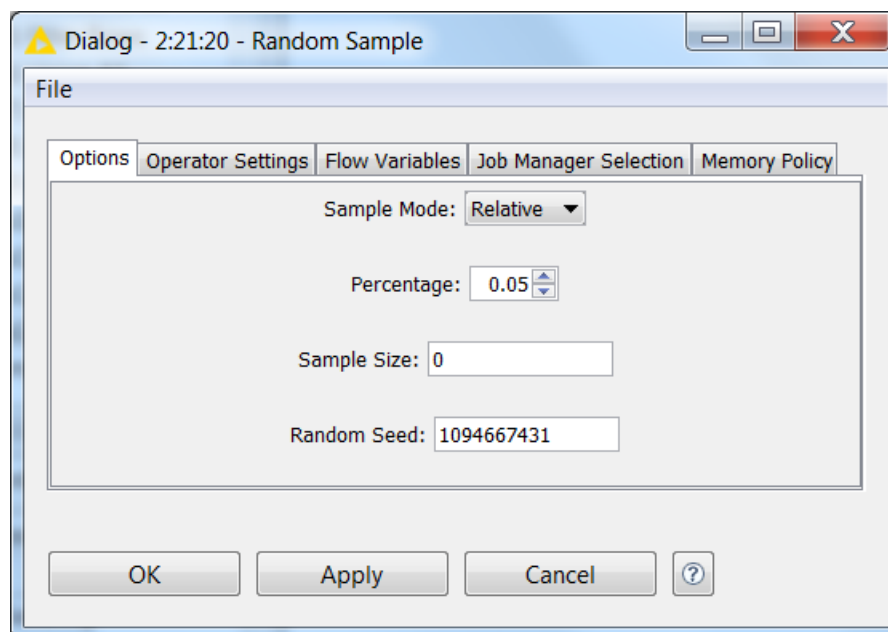
## Mixed Executor Workflow

DataFlow nodes include a basic set of analytics nodes, but virtually no visualization nodes. If you need any visualization more than a simple ROC curve, you will need to use KNIME non-flowable nodes to get that functionality. So, now is a good time to add non-flowable nodes, and learn how to mix the two executors in a single workflow.

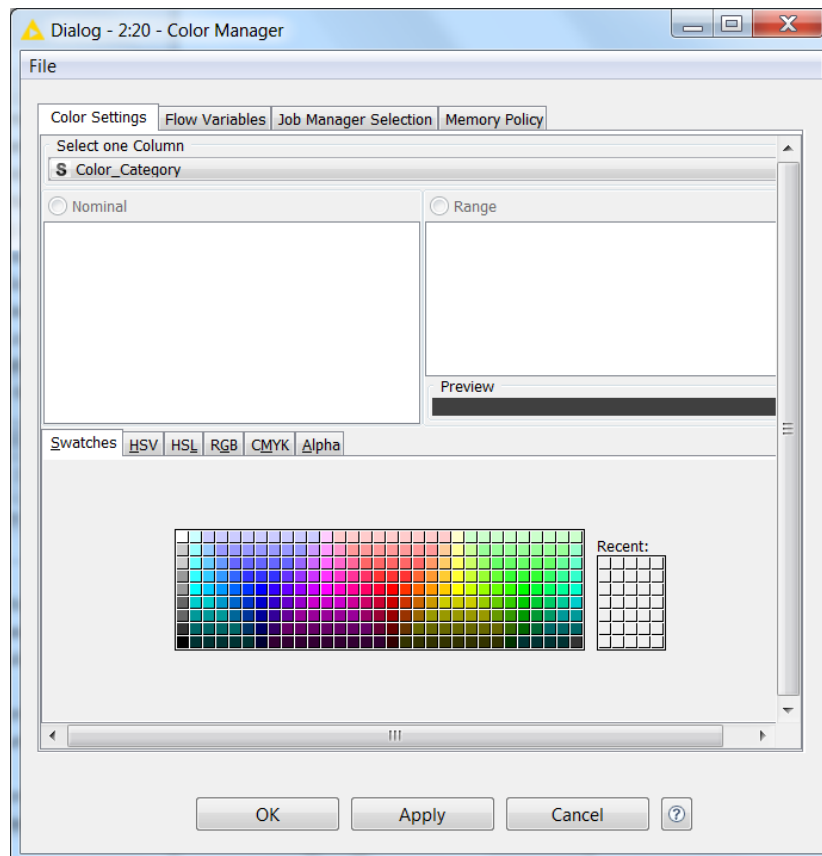
## Step 19: Add Data Visualization Nodes

Data visualization can only really display a certain number of data points. When dealing with large data sets, this is usually handled by displaying a sample of the data or summary data. Then, drill downs in BI software can allow examination of the data in greater detail. When using a simple visualization node, reducing the size of the data can give you faster performance without any degradation in the clarity of the visualization.

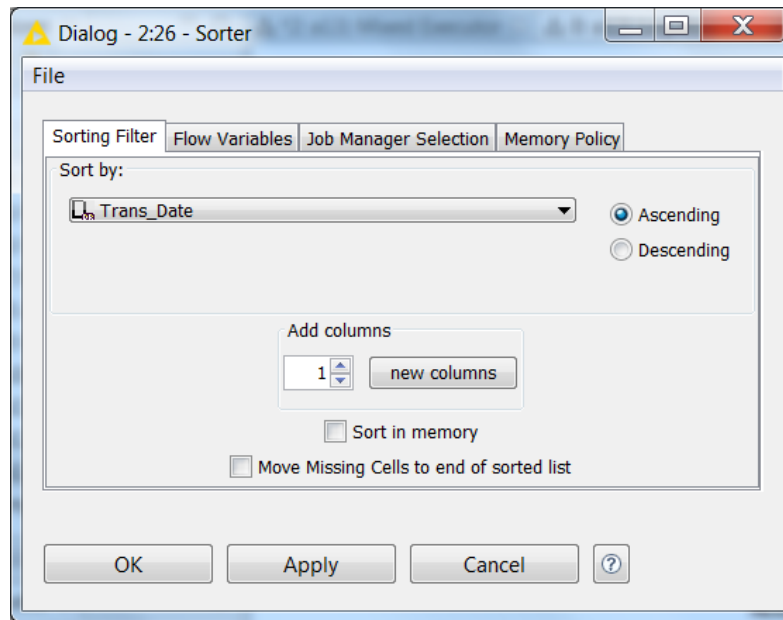
1. Add a Dataflow Random Sample node from under Transformation, Filter, and connect it to the output of the Run JavaScript node.
2. Set the Percentage to a 5% sample.



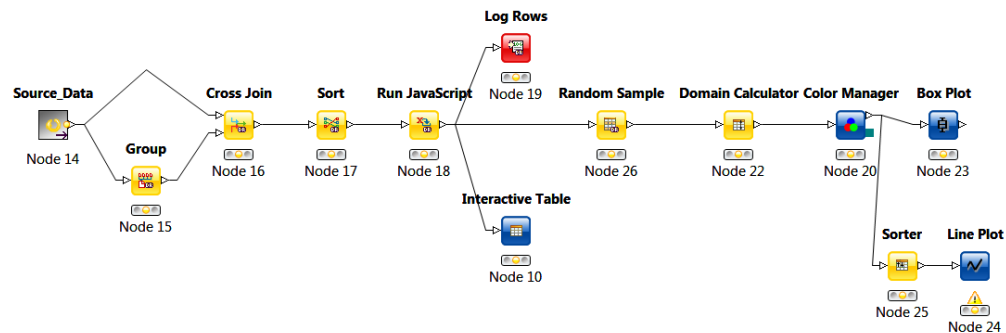
3. Now, we're going to start adding new nodes that are NOT DatFlow nodes. The first is a Domain Calculator node. Type domain in the search to find it. Connect it to the Random Sample node.
4. Click Add All to put all the fields into the visualization nodes to follow. In a practical situation, you would pick and choose which ones you wanted to visualize. In this case, we'll just grab them all.
5. Add a Color Manager node to the workflow and connect that to the Domain Calculator.
6. In the Color Manager configuration, under Select One Column, choose Color\_Category.



7. Add a Box Plot node and connect it to the top port, the data port, on the Color Manager. The Box Plot node doesn't require any configuration.
8. Next, place a row Sorter node in the workflow. Connect the Sorter to the Color Manager data port also. (Don't connect it to the Box Plot.)
9. Configure the Sorter to sort by Trans\_Date, Ascending only.



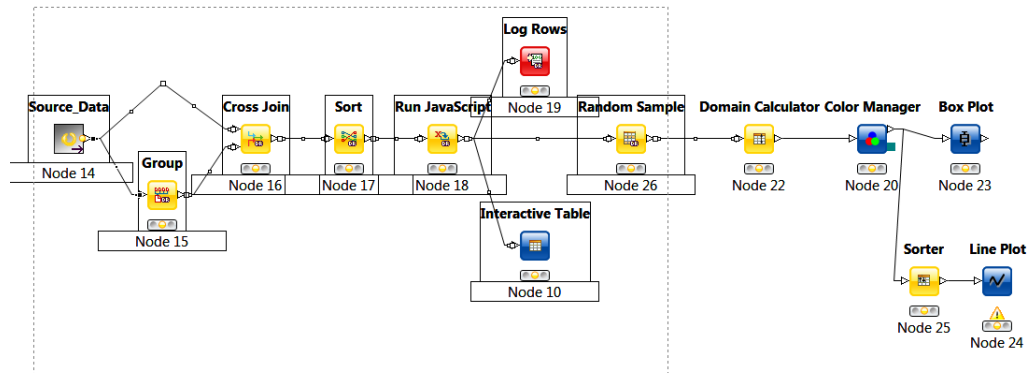
10. Next, add a Line Plot node, and leave the configuration set on the defaults.
11. Save your workflow, but **don't execute it yet**.



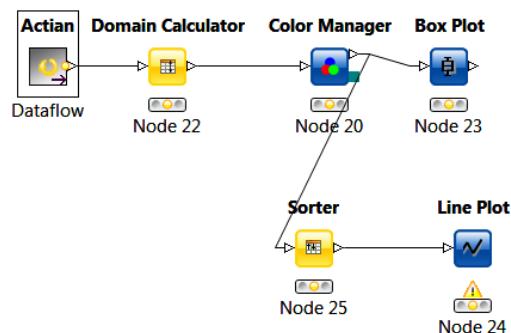
## Step 20: Separate Nodes by Type and Execute

Now, we have a workflow with half flowable nodes and half non-flowable nodes. Since flowable nodes can still use the default KNIME executor, this workflow will execute reasonably well using the default executor, but it will take quite a while. Since non-flowable nodes cannot reliably be executed using the DataFlow executor, this workflow cannot gain the parallel execution speed advantage. The best solution is to execute the flowable nodes with the DataFlow executor, then pass the data to the non-flowable nodes, and execute them with the default KNIME executor. To do this, we need to separate the two types of nodes.

1. Select all of the nodes at the beginning of the workflow up to, and including, the Random Sample node.



2. Collapse those nodes into a metanode, and name it something. I'll call mine Action DataFlow.
3. Right click and choose configure to get to the configuration window for the metanode. (You can't get it by double-clicking. That will just expand the metanode.)
4. On the Job Manager Selection tab, choose the DataFlow Executor. This is the one time when assigning the executor to a particular node is a good idea.
5. In the KNIME Explorer window on the left, right click on your workflow's name and choose Configure.
6. Change the Job Manager to the <<default>> KNIME executor.
7. Save your workflow. You now have all of your flowable nodes in one metanode set to use the DataFlow executor, and the rest of your workflow set to use the default KNIME executor. This should give you as much of an execution speed advantage as you can get for this workflow, and still give you all the functionality that you need. A good balance.
8. Execute your completed workflow.



**Save Point: Mixed Executor**



**Bonus Steps:** Explore the data visualizations

1. Right click on the Box Plot node and choose View: Box Plot

2. On the column selection tab, choose Account\_Number and click << remove. A plot of the account number makes no sense anyway, and now you can see the plots of the Balance and New\_Balance fields.
3. The Postcode doesn't make any sense on a box plot either, so remove it as well.
4. Because of the difference in scale between the other fields and the Deduction field, you can't see the plot for the Deduction field. Remove all the other fields until it is the only one left. Now, you can see the information for Deduction.
5. Close the Box Plot and have a look at the Line Plot.
6. It is very difficult to make sense of the data. Remove the Account\_Number and Postcode. Now, you can see a clear pattern in the balance and deductions. The Color Legend tab will help you see which line is which.
7. On the Color Legend tab, click Change next to Balance.
8. Choose a different color that will be easier for you to distinguish, then click OK.



**Tip:** KNIME has a lot of useful visualization operators. OpenText (formerly Actuate) also provides an open source visualization node pack for KNIME called BIRT which you can download and add to your KNIME workbench.



**Tip:** If you want sophisticated interactive dashboards, you may wish to feed your analytic results into a BI visualization tool such as Tableau, Yellowfin, MicroStrategy or the OpenText commercial version. To do this, you can pass the data to these other applications from KNIME by putting a writer node at the end of your workflow. Write to a database or file format that those applications can read or integrate with directly. For example, Actian's free Vector Express or Vortex Express databases will integrate nicely with all of those I named. Naturally, our DataFlow operators write to those databases very efficiently via the Load Action Vector or Load Action Vector on Hadoop nodes. You can also write to open source options like MySQL or simply write to a file format like Delimited Text. The DataFlow Database Writer node will write to any database with a JDBC connectivity option.



**Bonus Steps:** Execute the workflow with the KNIME default executor.

1. Right click on the Actian DataFlow metanode and expand it.
2. Expand the Source Data metanode as well. Since the two metanodes no longer exist, the setting to make them run with the DataFlow executor also no longer exists.
3. The workflow should already be set to use the KNIME default executor over all. Execute the workflow and watch the progress. Note the differences in how the KNIME default executor and the DataFlow executors work.
4. After the workflow has completed executing, right click on the first node in the workflow, or on the workflow name in the KNIME Explorer and choose Reset. This will get the workflow ready to be executed again.

5. Right click on a node in the middle of the workflow somewhere, such as the Filter Rows node. Choose Execute.
6. Watch the workflow. Notice that it only executes up to the node that you clicked on. The rest of the workflow does not execute. This is one of the things that the default KNIME executor can do that the DataFlow executor cannot do. The default KNIME executor can execute a single node at a time, or execute the entire workflow up to a particular node.



**Tip:** When de-bugging and testing a workflow, and trying to pinpoint problem areas, use the KNIME executor. Because the KNIME executor executes each node action on the full data set before moving to the next node action in the workflow, you can check data sets as you go along and spot which node is having a problem more easily. DataFlow's pipeline execution model makes a workflow an all-or-nothing kind of procedure. Starting the workflow execution is like turning on the water tap at the front of a water hose. This makes the KNIME default executor far more useful for stepping through a workflow to spot problems.



Actian is delighted to provide the DataFlow operators and executor free of charge to the KNIME community. We hope it helps you get your analytics work done much more efficiently.

Please, join the Actian DataFlow community and let us know if you have any problems, or if you have a cool tip or trick to share. <http://supportservices.actian.com/community>

If you find that you need more data processing power than the free version of DataFlow provides, please contact us at [info@actian.com](mailto:info@actian.com) or [sales@actian.com](mailto:sales@actian.com) for information on the commercial version.

If you need to add a high performance, low-latency SQL database to your analytics job, please try our Actian Vector™ Express free single server database, or our Actian Vortex™ Express (Vector in Hadoop) free Hadoop database, both of which come pre-packaged with KNIME and DataFlow. Vector Express can handle up to 250 GB of data and Vortex Express can handle up to 500 GB of data.

All Actian free downloads are available here: <http://www.actian.com/product-downloads/>

If you need fast queries on more data than 250 or 500 GB, contact [info@actian.com](mailto:info@actian.com) or [sales@actian.com](mailto:sales@actian.com) for information on the commercial versions of Actian Vector and Actian Vortex.