ELTMaestro User Guide for Spark

ELTMaestro

Version 11.0.0

IQ Associates Inc.

Contents

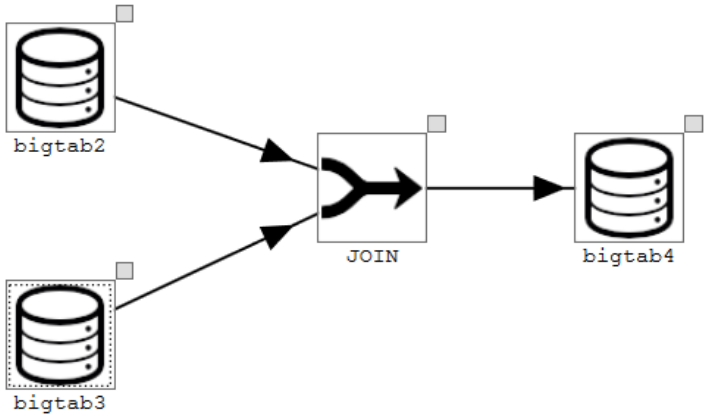
Introduction

ELTMaestro is a tool that helps you do data integration and build data warehouses on powerful platforms such as IBM PureData for Analytics (Netezza), Amazon RedShift, Apache Spark, and other systems. This manual covers the use of ELTMaestro on Apache Spark.

ELTMaestro Concepts

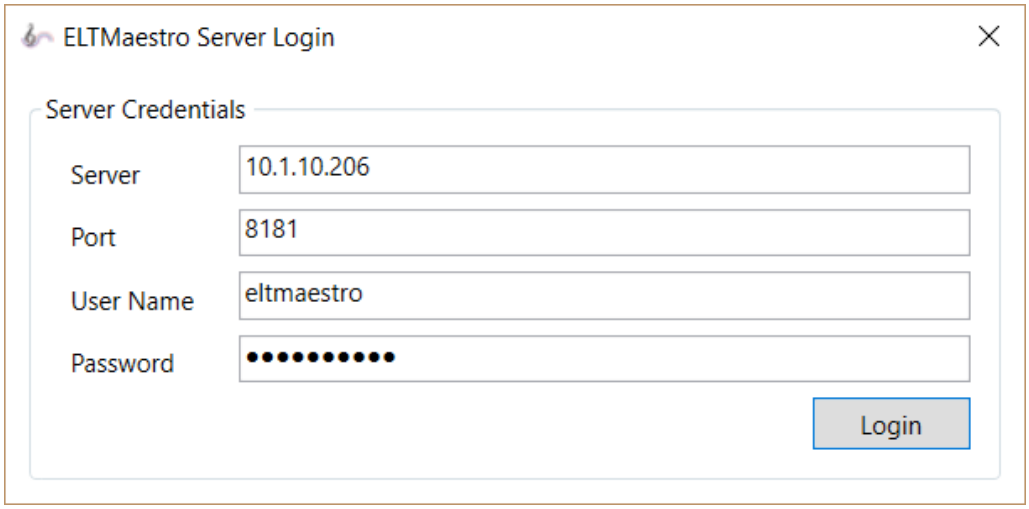
To understand this document and effectively employ the ELTMaestro, it is first necessary to define a few terms and concepts frequently used within the data integration community. These terms are used differently by different data integration tool vendors, so it is best to establish our terminology at the outset.

- A ***job*** is a reusable component that does some defined work. A single job may read from one or more data sources and write to one or more targets. In most cases, the processing within a job is atomic and consistent, meaning the job as a whole will either completely succeed or fail. Data will either be committed or rolled back to its pre-job state based on job success or failure. In either case, the ELTMaestro engine will clean up and remove all temporary tables and other intermediate processing artifacts.
- A ***batch*** is a collection of jobs that are run together. The batch defines the order in which the component jobs are run. Jobs within a batch are connected by a graph which illustrates how control flow passes from job to job, as shown below.
- In this manual we use the term ***workflow*** to refer to jobs or batches. A workflow can execute multiple network of workflows and can have infinite hierarchy.
- Just as jobs are the components of batches, ***steps*** are the components of jobs. Consider the following example: A job might extract data from two tables, combine the two datasets, massage the data into the form required by a target table, and then load the data to the target. Such a job might be composed of the following operations: (1) Extract the data from the first source, (2) extract data from the second source, (3) join the two datasets, and (4) load the resulting dataset into the target table. Such a job would be composed of four *steps,* as shown below.



Starting the ELTMaestro Client

When you launch the ELTMaestro Data Integration Client, the Login screen appears as shown below:



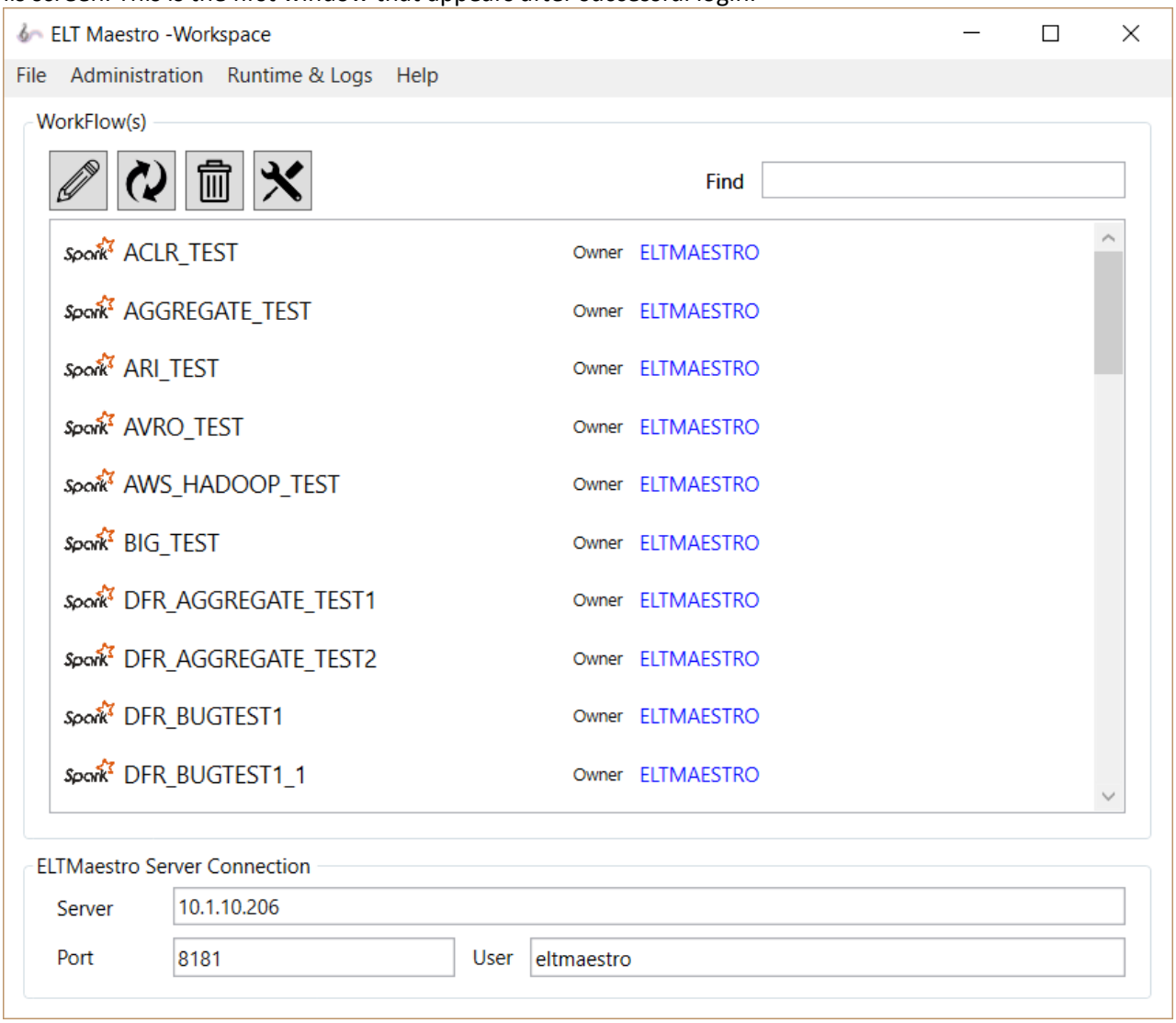
Parameters:

Property	**Type**	**Info**
Server	Text	Hostname or ip address of ELTMaestro server.
Port	Text	ELTMaestro server messaging port. Default is 8181.
User Name	Text	ELTMaestro data integration username. The credentials are not affiliated with other accounts.
Password	Text	ELTMaestro data integration user password.

Click [Login] after populating required credentials. Upon successful login the Workspace Window should appear. You can then start creating workflow jobs and build dataflow diagrams.

The Workspace

All created jobs loaded from server workspace folder appear on this screen. Workflow jobs can be created or edited from this screen. This is the first window that appears after successful login.



Parameters:

Property	Type	Info
Edit	<double click on job>	Opens job editor for selected workflow job.
Create New Workflow	Button	Creates a new workflow job.
File->Create Work Flow	Menu	Creates a new workflow job.
Delete	Button	Deletes selected workflow job.
Find	Search Box	Workflow job keyword search
Refresh	Button	Refresh workflow jobs list
Set Job Configuration	Button	Opens configuration window for currently selected job.

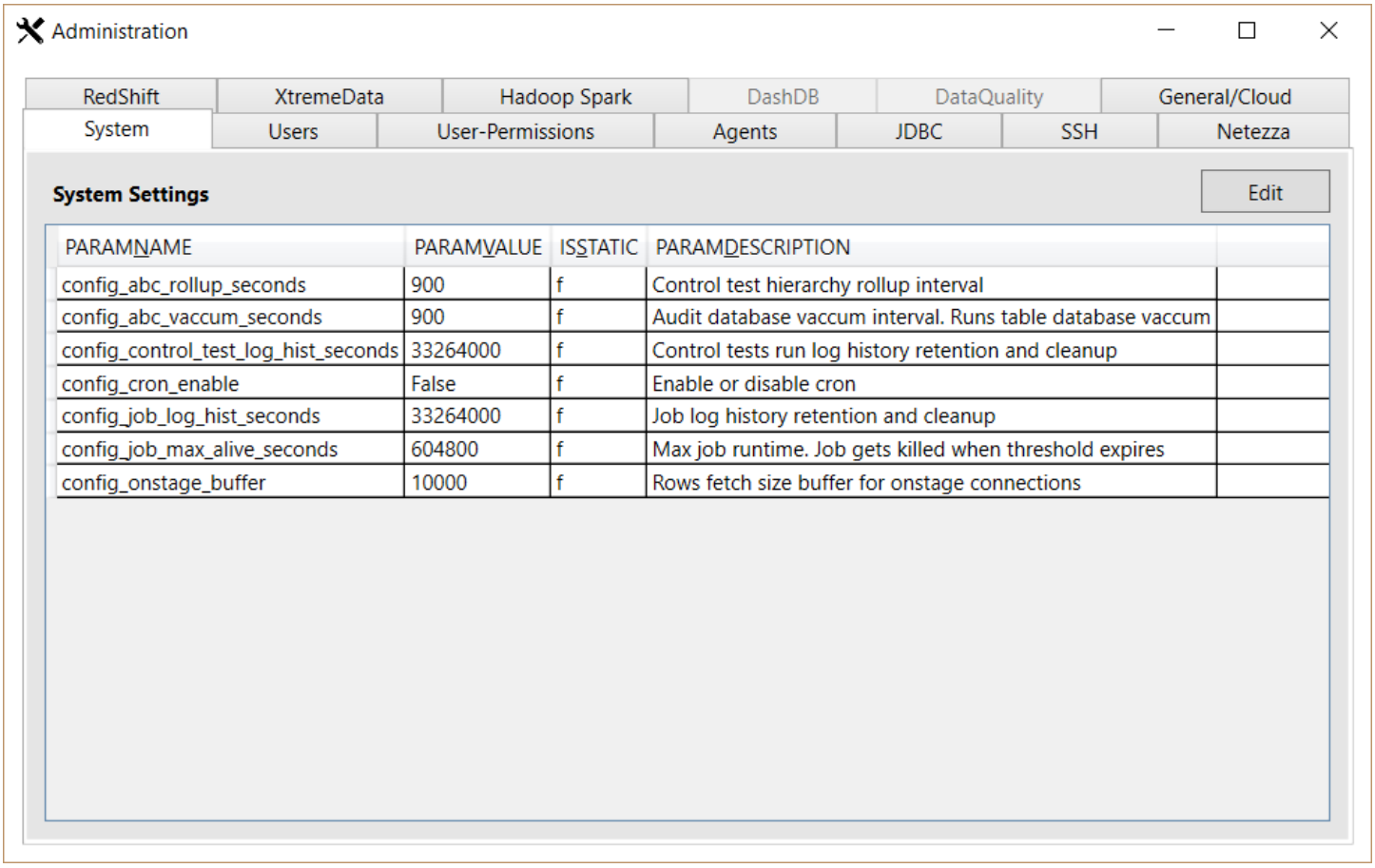
Administration ->Configure Maestro Server	Menu	Opens ELTMaestro server configuration window.
Administration ->Scheduler	Menu	Opens Job Scheduler manager. Crontab Editor.
Runtime & Logs ->View Logs	Menu	Opens Log Viewer.

Configuring ELTMaestro Server

On Workspace Window menu, Click [Administration] -> [Configure Maestro Server].

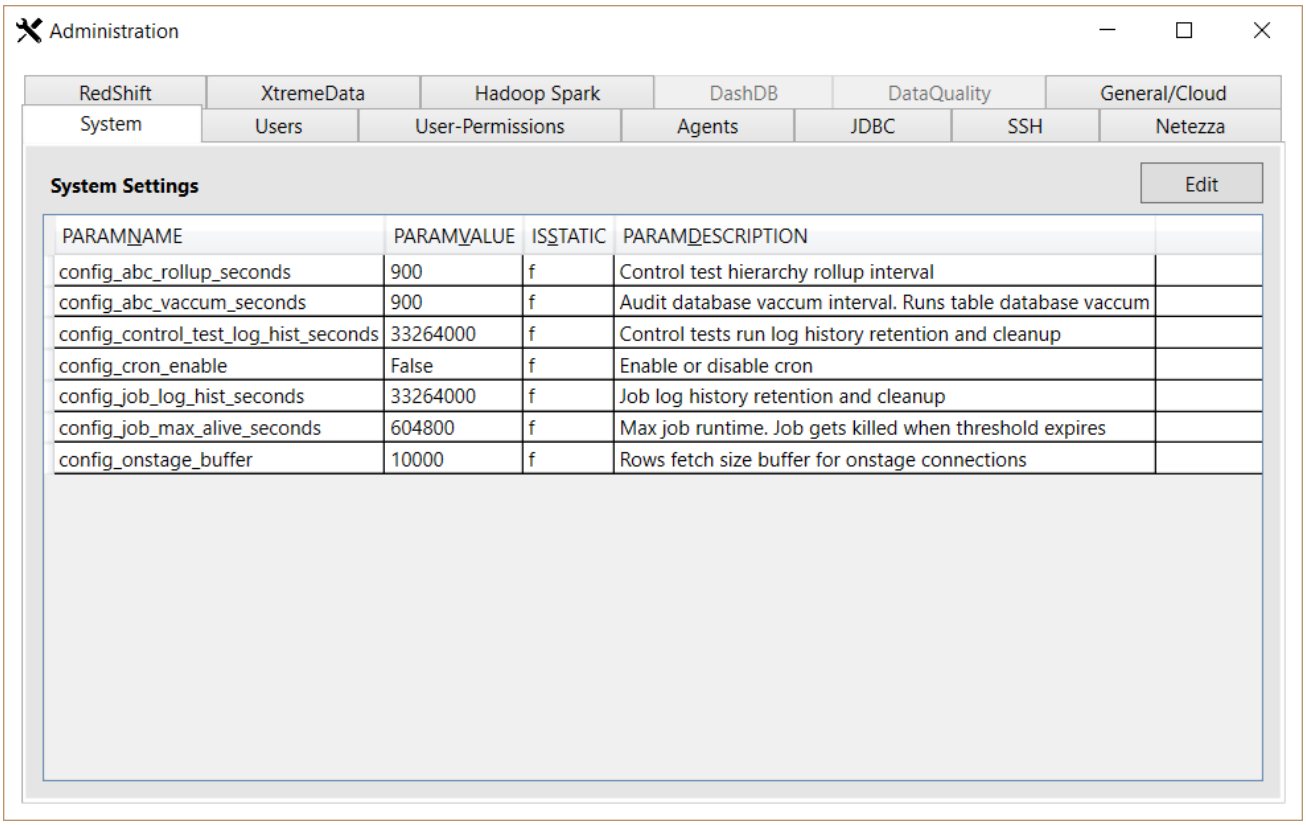
System Settings

System tab contains default ELTMaestro engine configurations. Generally, these settings are left unchanged.

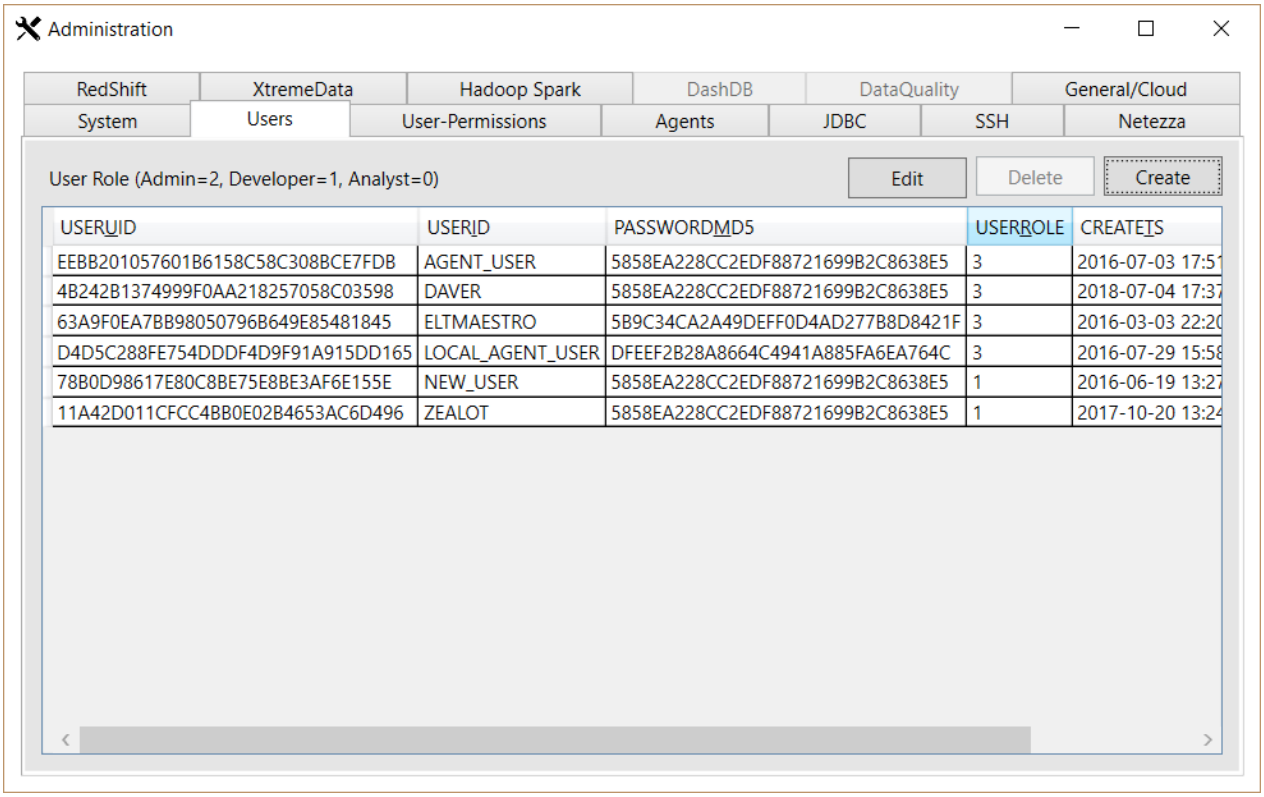


Users Settings

Create or modify new ELTMaestro users in this tab. There are three different roles that are assigned to users as shown below. For system users or service accounts (ELTMaestro Agents use), only create/use users with role level=3 (system). It is not recommended to use system user credentials on ELTMaestro Client Application. Accounts also gets locked after 10 failed attempts. Change the default password here by clicking [Edit].



PARAMNAME	PARAMVALUE	ISSTATIC	PARAMDESCRIPTION
config_abc_rollup_seconds	900	f	Control test hierarchy rollup interval
config_abc_vaccum_seconds	900	f	Audit database vaccum interval. Runs table database vaccum
config_control_test_log_hist_seconds	33264000	f	Control tests run log history retention and cleanup
config_cron_enable	False	f	Enable or disable cron
config_job_log_hist_seconds	33264000	f	Job log history retention and cleanup
config_job_max_alive_seconds	604800	f	Max job runtime. Job gets killed when threshold expires
config_onstage_buffer	10000	f	Rows fetch size buffer for onstage connections

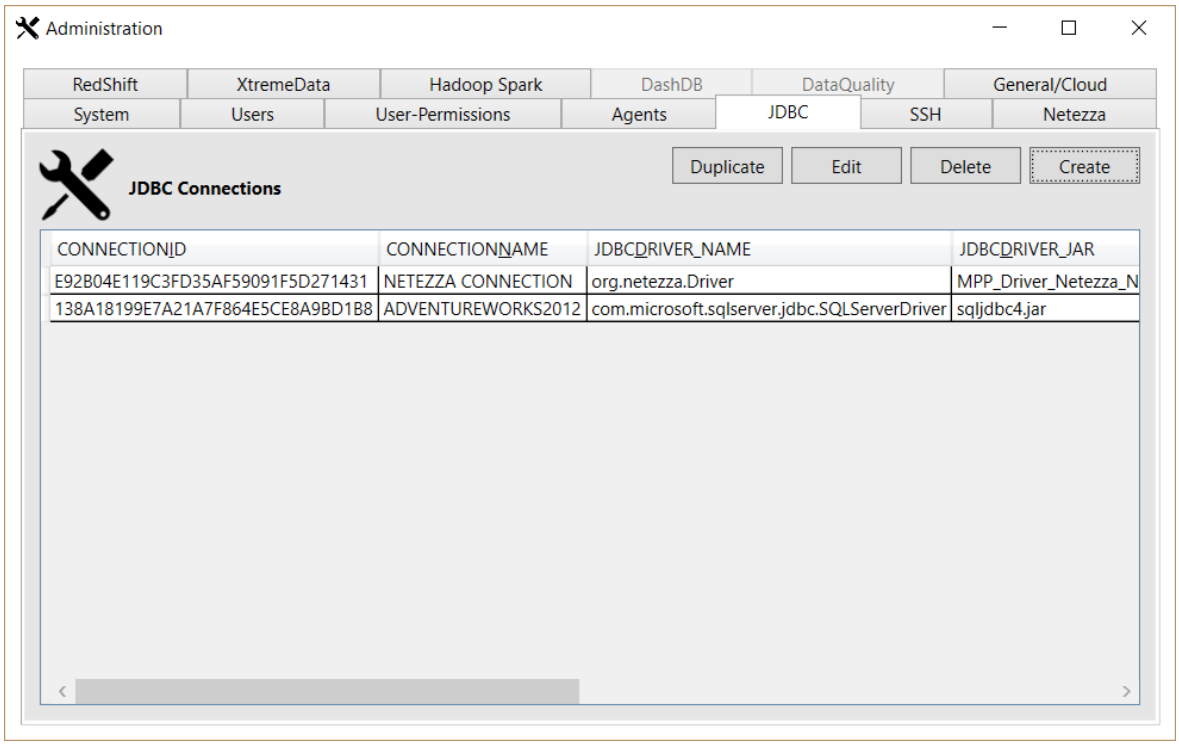


Creating/Modifying User

On Users Tab Click [Create] and assign necessary credential properties then Click [Apply]. To modify user credentials, select a user, Click [Edit], make any necessary changes then Click [Apply]. Newly created user can then be used on Client Application login.

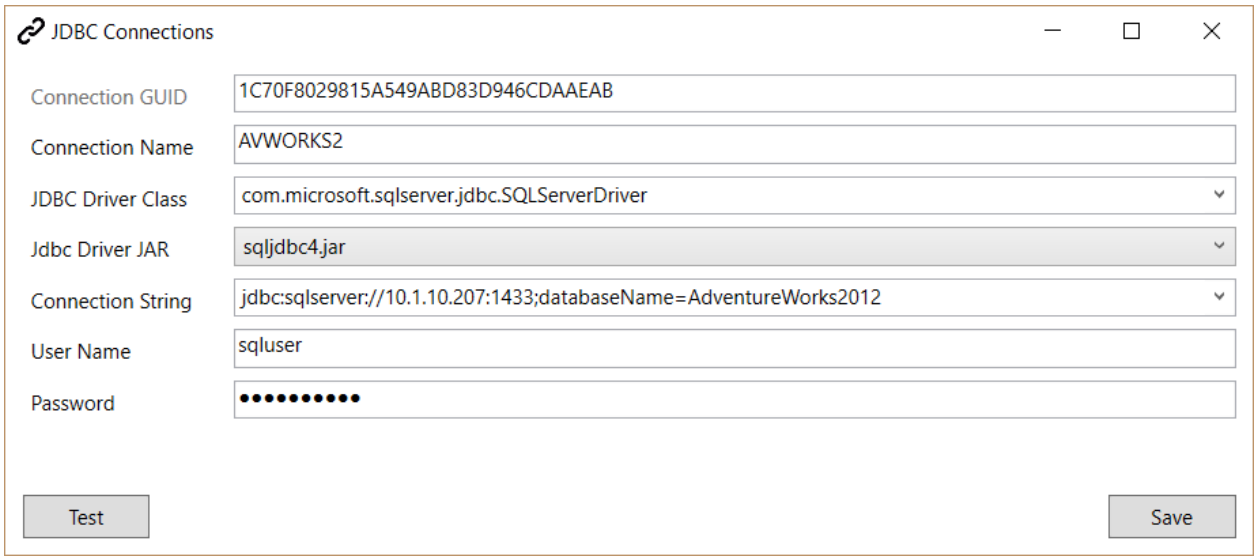
JDBC Settings

This configuration window contains all available relational database connection settings including connection setting for targeted MPP platform.



Creating JDBC Connection Property

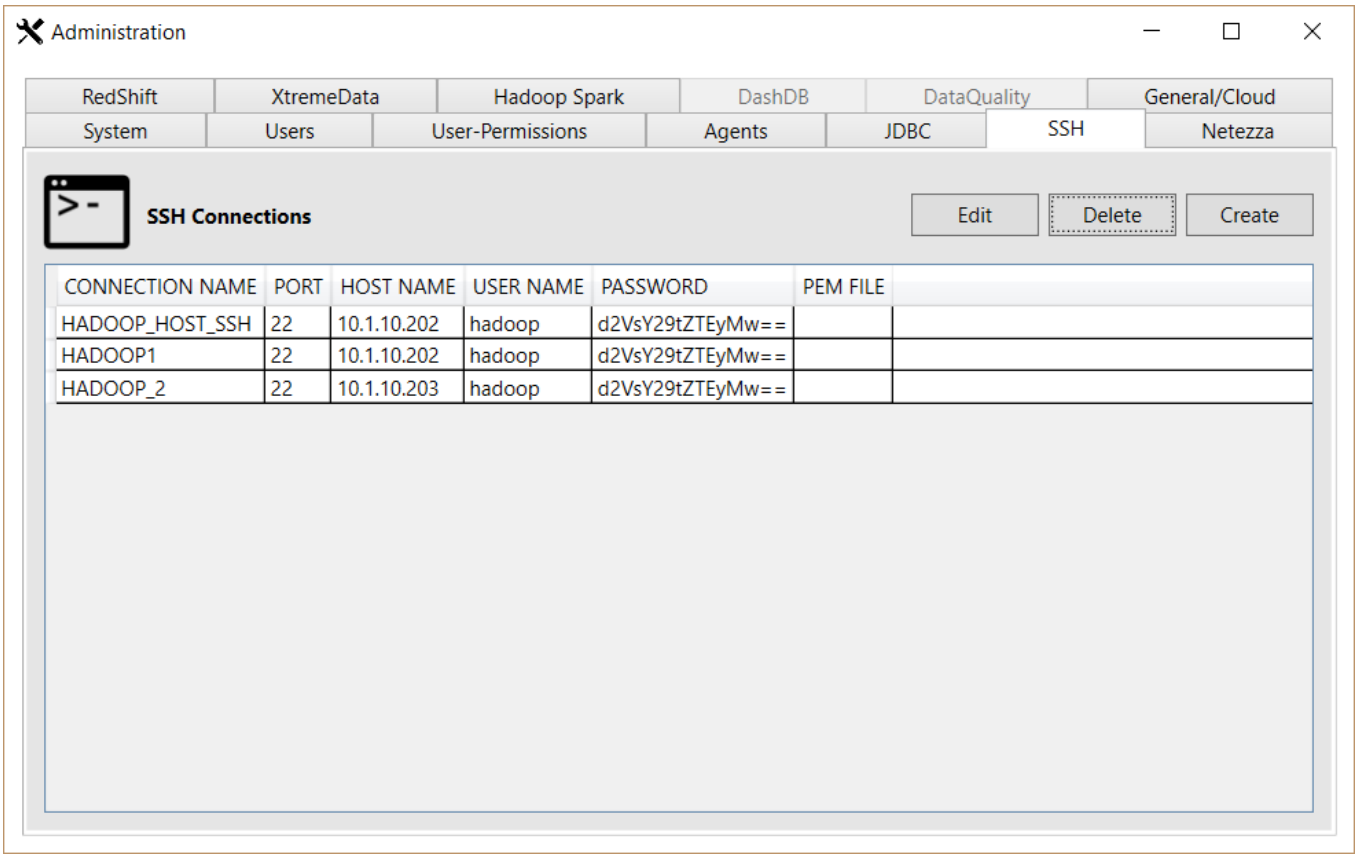
To create a new JDBC connection, Click [Create]. Connection property window appears as shown below. Fill in the details. Click [Test Local] button to test connectivity from ELTMaestro server.



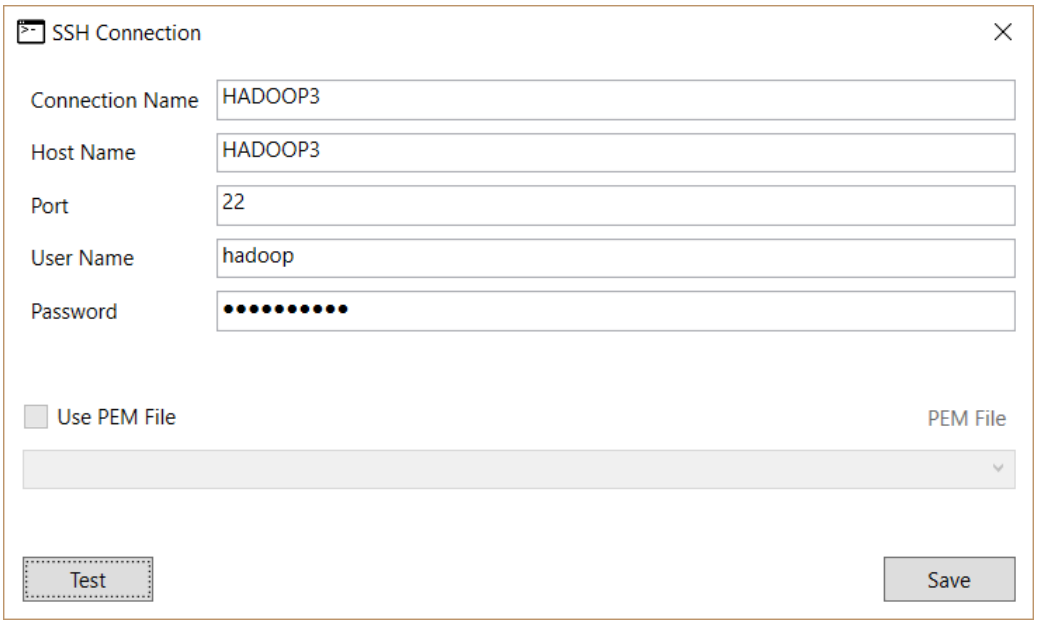
Click [Save] when done.

SSH

SSH Connections are used by SSH and SFTP steps.



SSH Property for creating/editing as shown. Click [Create] to create new SSH connection or select a connection and Click [Edit] to modify existing connection.

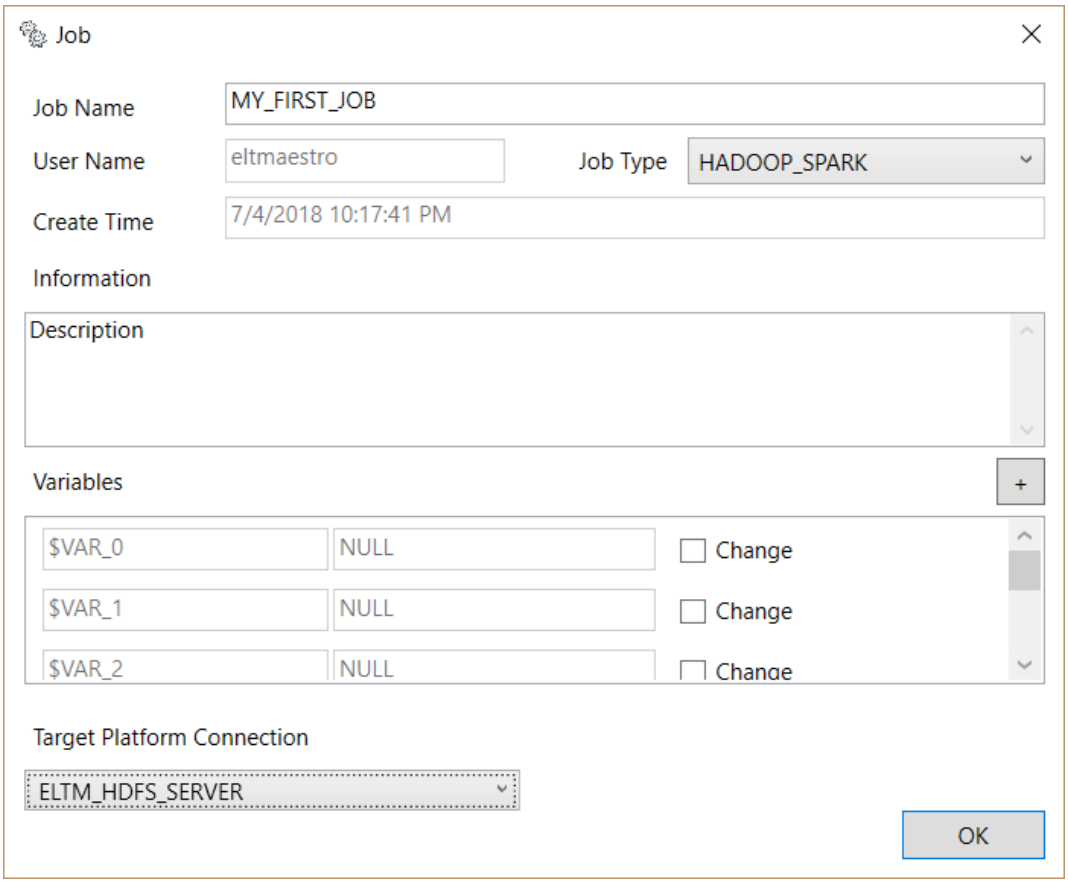

SSH Connection
Connection Name
HADOOP3
Host Name
HADOOP3
Port
22
User Name
hadoop
Password
Use PEM File
PEM File
Test
Save

Working with Jobs (Workflow)

Creating a New Job

Most of ELT mapping design happens on a workflow job. Jobs contain steps and mapping lines. Steps are the smallest purpose-driven objects. Mapping lines link the steps together, control parallelism, order data flow and map columns.

Clicking on the [New] button in the Workspace brings up the *Job Creation* dialog:



Job Variables

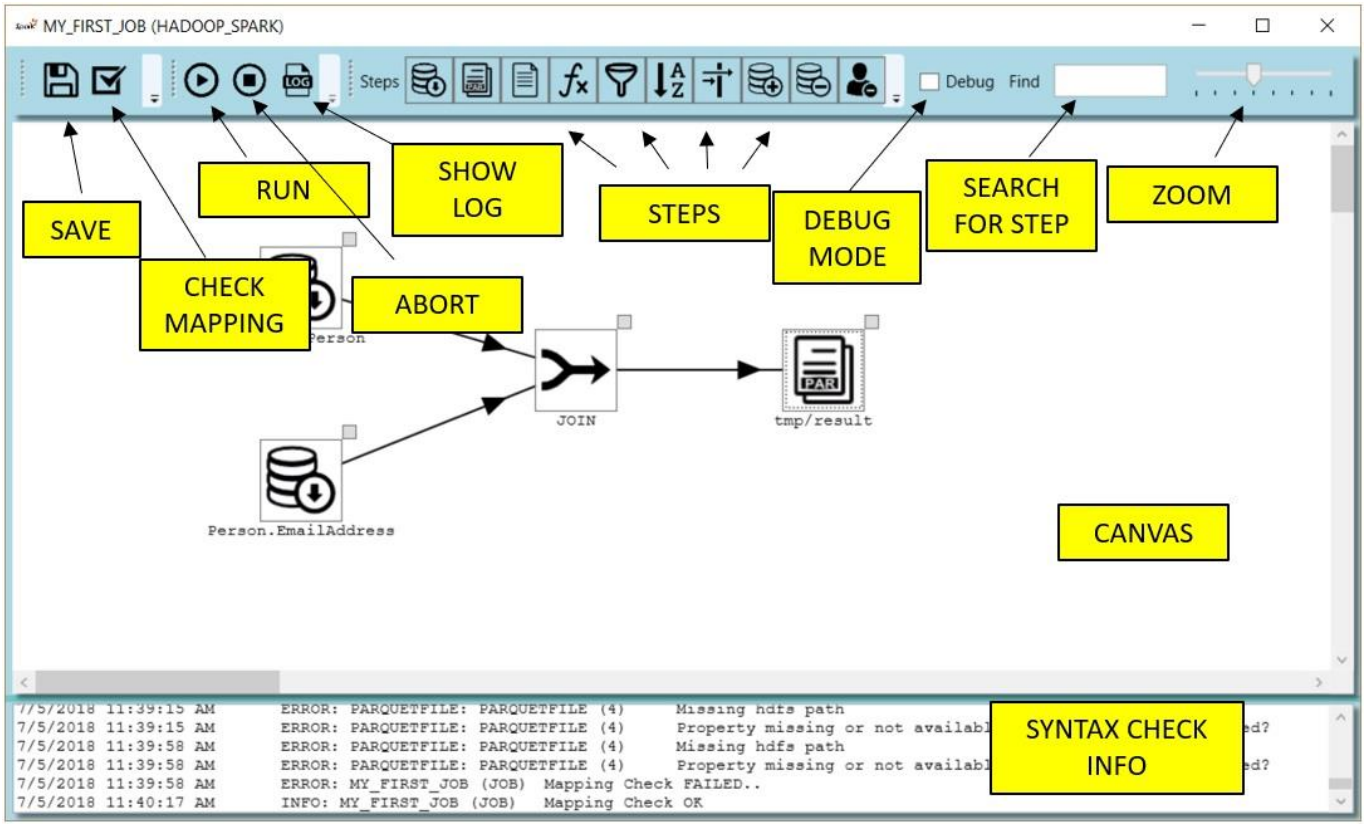
Each job is associated, by default, with 5 job variables, named $VAR_0, $VAR_1, $VAR_2, $VAR_3, and $VAR_4. Job variables are used to pass information between ELTMaestro programs and external databases or Unix shells, or between one part of the program and another. If you think you need more than 5 job variables, you can add more by clicking the "+" button. You can add as many as you like. By default, job variables are initialized to NULL; to initialize to another value, click the "Change" box next to the variable.

Once a job is created the variables cannot be added or removed. Job variables are discussed further below, in the discussion of the "Set Variable" step.

Editing a Workflow Job

- On Workflow widow select job name.
- Click [Edit] or [Double Click] selected job.

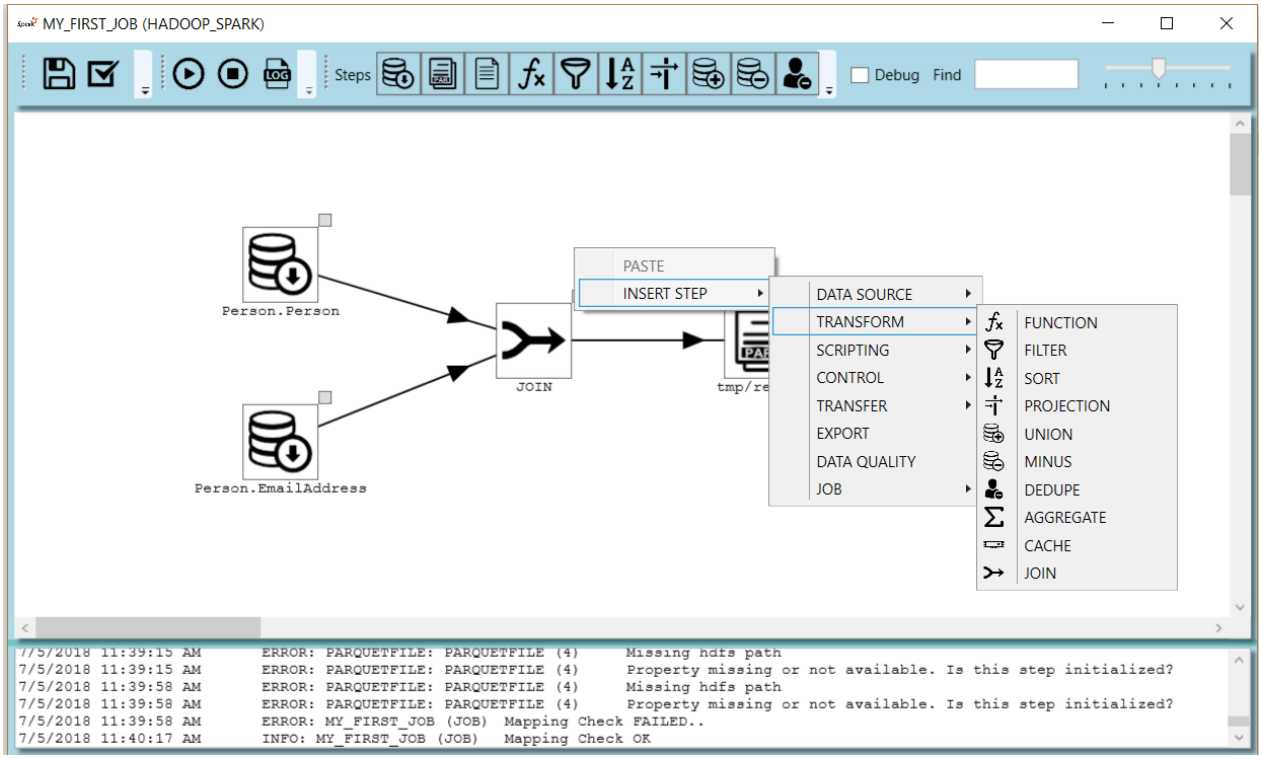
Job Interface Overview



Adding a step

To add a step in a workflow

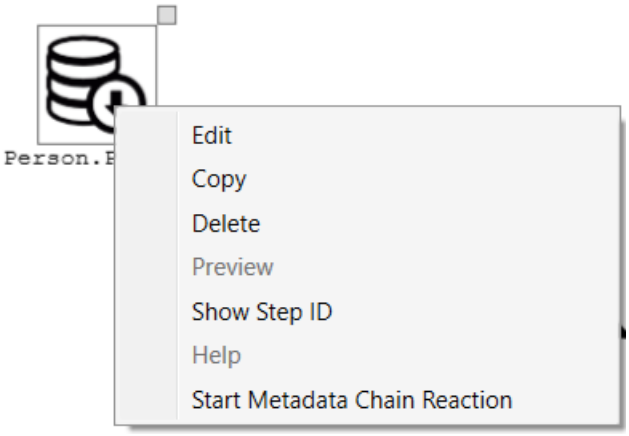
- [Right-Click] on canvas
- Select step to be added



Steps can also be drag-dropped into canvas from menu bar.

Removing a step

- [Right-Click] on step on canvas
- Click [Delete]

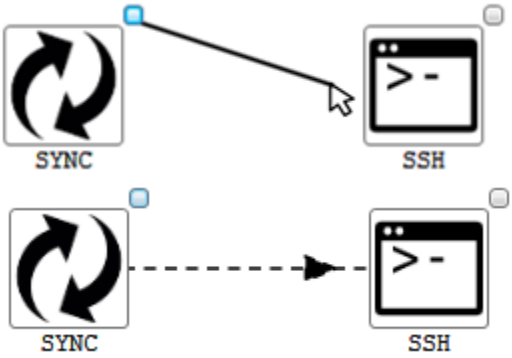


Editing Step

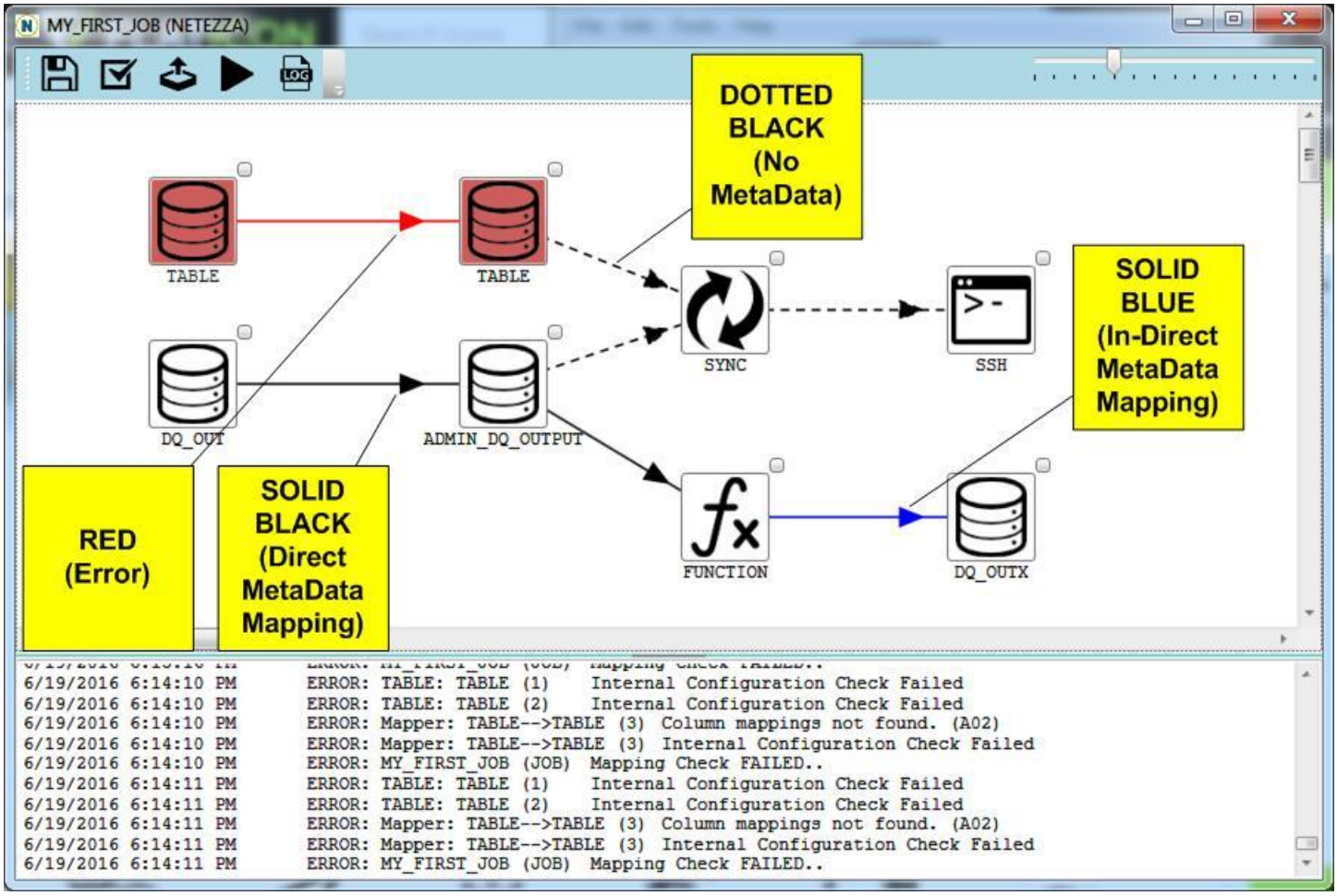
- [Right-Click] on step
- Click [Edit]
- OR [Double-Click] on step

Adding Flow Line (Mapper)

- Click on source step [button] (on top right) without releasing mouse button.
- While holding mouse button move cursor on target step and release.



Flow line connects one or more steps. The purpose varies depending on the types of steps connected.



Editing Mapper

- Click on the arrow head to open mapper properties dialog.

The Mapper properties are not available if there is no metadata (or a dotted line).

Spark ELT Steps

Step Types

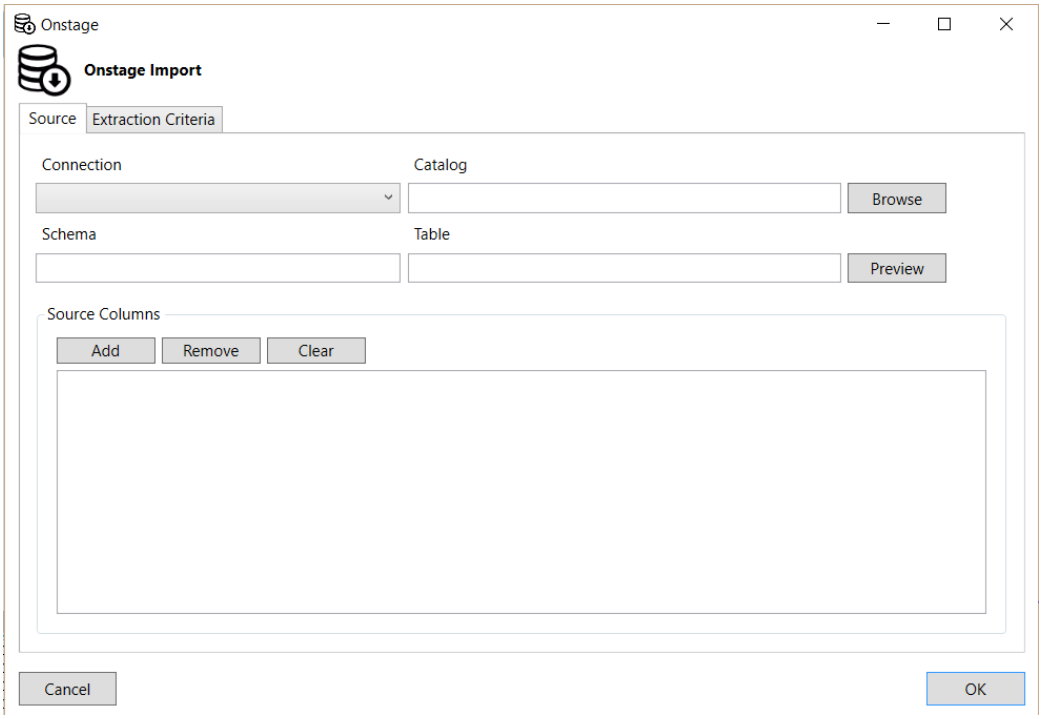
	Step Type	Group	Information
	Onstage	Data Source	Relational databases as data source utilizing JDBC connection.
	Parquet File	Data source	I/O for Parquet files
	File Reader	Data Source	I/O for structured delimited file.
	Function	Transform	Database supported scalar function transformation.
	Filter	Transform	Filter or restrict data based on conditions.
	Projection	Transform	Remove columns.
	Union	Transform	Union two input sources.
	Minus	Transform	Diff two input sources.
	Dedupe	Transform	Remove duplicates.
	Aggregate	Transform	Apply general aggregate functions.
	Join	Transform	Join two input sources.
	SQL Script	Scripting	Execute SQL Script. Insert, update, delete, truncate, drop statements.
	SSH	Scripting	Execute shell script.
	Sync	Control	Control step parallelism. Dummy step.
	Switch	Control	Branch success/failure paths based on input step state.
	FileWatch	Control	Watch for file availability.
	JDBCWatch	Control	Watch for value availability with SQL Query.
	SetVariable	Control	Set variable value for variables defined while creating job.
	WaterMark	Control	Set current job watermark. Set root job watermark.
	SFTP2S3	Transfer	Pull files into ELTMaestro server from SFTP server.
	JobStep	Job	Run deployed job under current workflow.

OnStage

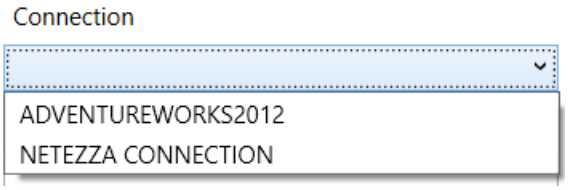
Introduction

The OnStage step is used to import relational data from sources outside the target system.

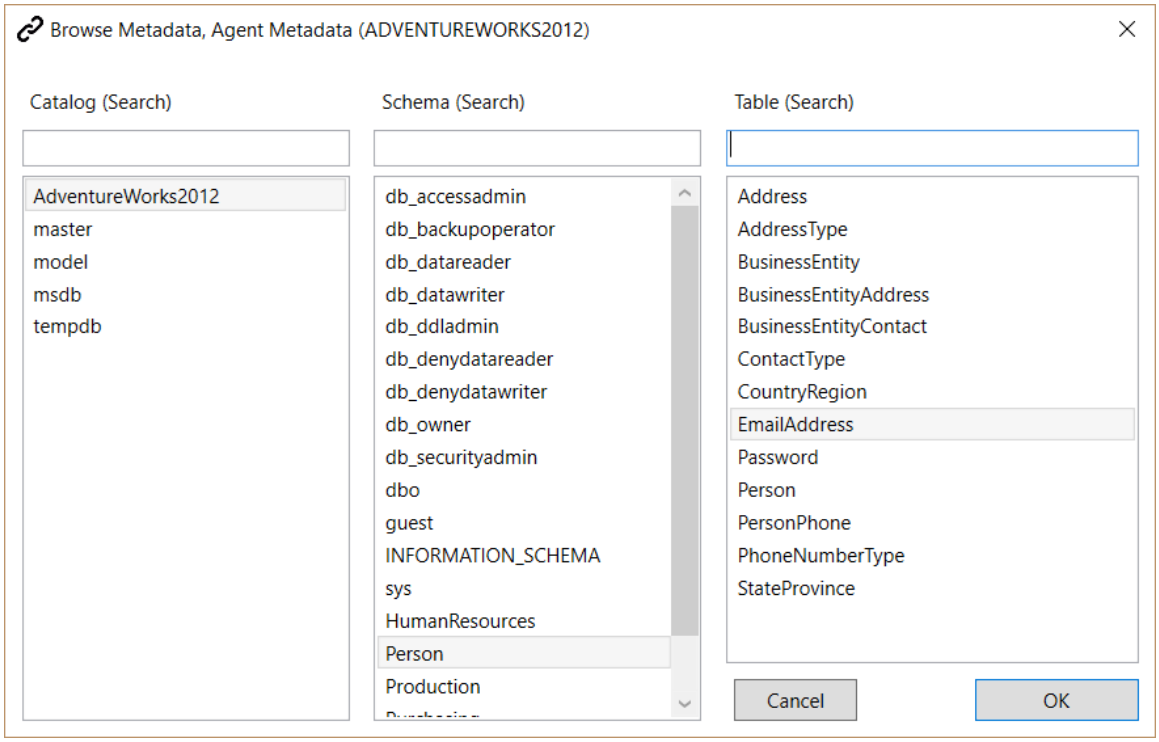
Example



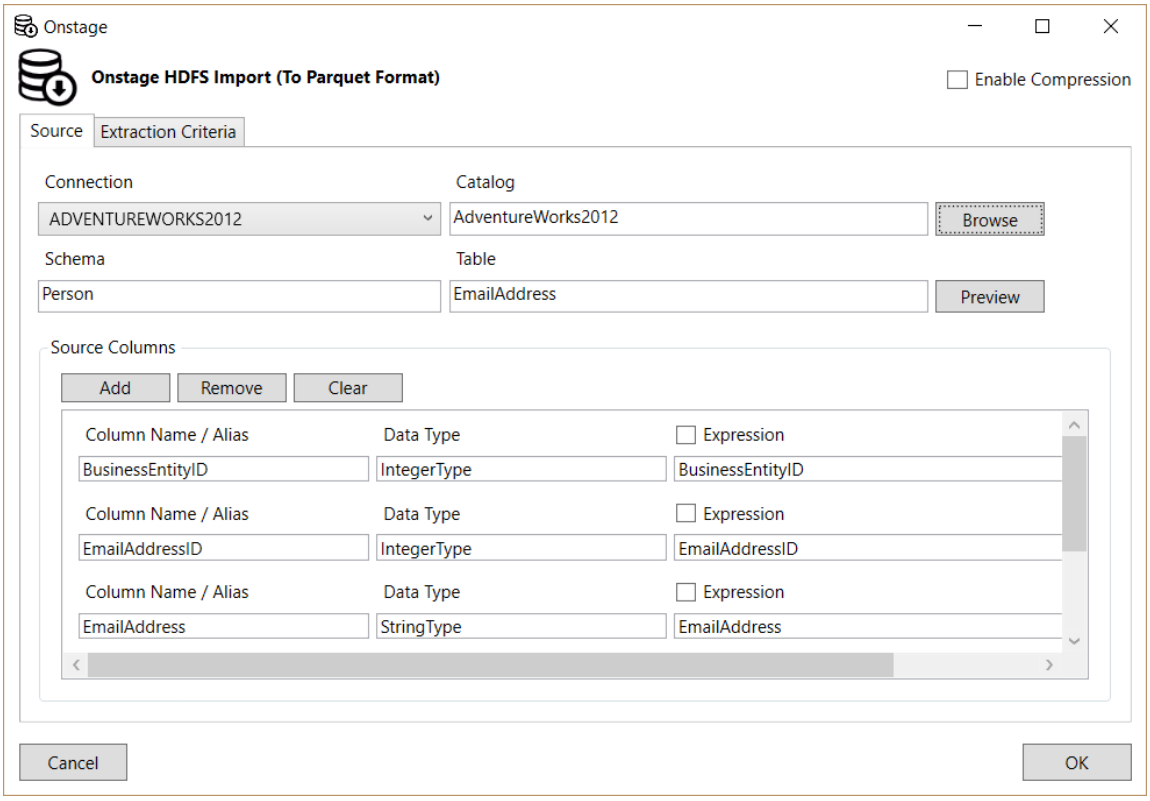
Note the *Connection* drop-down list. The *Connection* drop-down list allows the user to choose among sources that are visible to the ELTMaestro server.



After choosing a source, browse the catalogs to choose a catalog (i.e. database), schema, and database table:



and hit OK. You will be returned to the OnStage step properties window, with the metadata filled in:



The Preview button allows you to preview the contents of the table:

Preview

BusinessEntityID	EmailAddressID	EmailAddress	rowguid	ModifiedDate
1	1	ken0@adventure-works.com	8A1901E4-671B-431A-871C-EADB2942E9EE	2003-02-08 00:00:00.0
2	2	terri0@adventure-works.com	B5FF9EFD-72A2-4F87-830B-F338FDD4D162	2002-02-24 00:00:00.0
3	3	roberto0@adventure-works.com	C8A51084-1C03-4C58-A8B3-55854AE7C499	2001-12-05 00:00:00.0
4	4	rob0@adventure-works.com	17703ED1-0031-4B4A-AFD2-77487A556B3B	2001-12-29 00:00:00.0
5	5	gail0@adventure-works.com	E76D2EA3-08E5-409C-BBE2-5DD1CDF89A3B	2002-01-30 00:00:00.0
6	6	jossef0@adventure-works.com	A9C4093A-4F4A-4CAD-BBB4-2C4E920BACCB	2002-02-17 00:00:00.0
7	7	dylan0@adventure-works.com	70429DE4-C3BF-4F19-A00A-E976C8017FB3	2003-03-05 00:00:00.0
8	8	diane1@adventure-works.com	37F02A87-058D-49F8-A20D-965738B0A71F	2003-01-23 00:00:00.0
9	9	gigi0@adventure-works.com	F888A16D-0C33-459E-9D72-D16AE0BB1F43	2003-02-10 00:00:00.0
10	10	michael6@adventure-works.com	E0DD366D-433D-4F5A-9347-1A5FE7FBE0A3	2003-05-28 00:00:00.0
11	11	ovidiu0@adventure-works.com	0FF9523D-F398-4237-85F8-2834DE441692	2004-12-29 00:00:00.0
12	12	thierry0@adventure-works.com	B2962849-CC5F-4E57-BCB4-019642BBD8ED	2002-01-04 00:00:00.0
13	13	janice0@adventure-works.com	64871268-3812-402F-8A91-C618B6515B06	2005-01-16 00:00:00.0
14	14	michael8@adventure-works.com	BEA9075C-1BED-4E5E-8234-F5641FAF814C	2005-01-23 00:00:00.0
15	15	sharon0@adventure-works.com	5CD782BA-F5AB-41EC-B206-09B06F52C96B	2005-02-11 00:00:00.0
16	16	david0@adventure-works.com	80C0D44A-78B6-42AC-AE8D-410A22A6BE63	2002-01-13 00:00:00.0
17	17	kevin0@adventure-works.com	F4332215-C861-4A54-99F5-B0C4F6406515	2001-02-19 00:00:00.0
18	18	john5@adventure-works.com	A944FC67-F16E-4AB8-8940-8CEFF186BED3	2005-03-03 00:00:00.0
19	19	mary2@adventure-works.com	626CB0E6-D6A9-4A73-8227-ACB02DBC5214	2005-03-10 00:00:00.0
20	20	wanida0@adventure-works.com	924A0FC3-F9E4-4100-8C82-1C69AAE4501F	2005-01-31 00:00:00.0

OK

Using the Extraction Criteria tab, you can set conditions on what rows are extracted from the source tables:

Expression

EXPRESSION

Expression Attributes

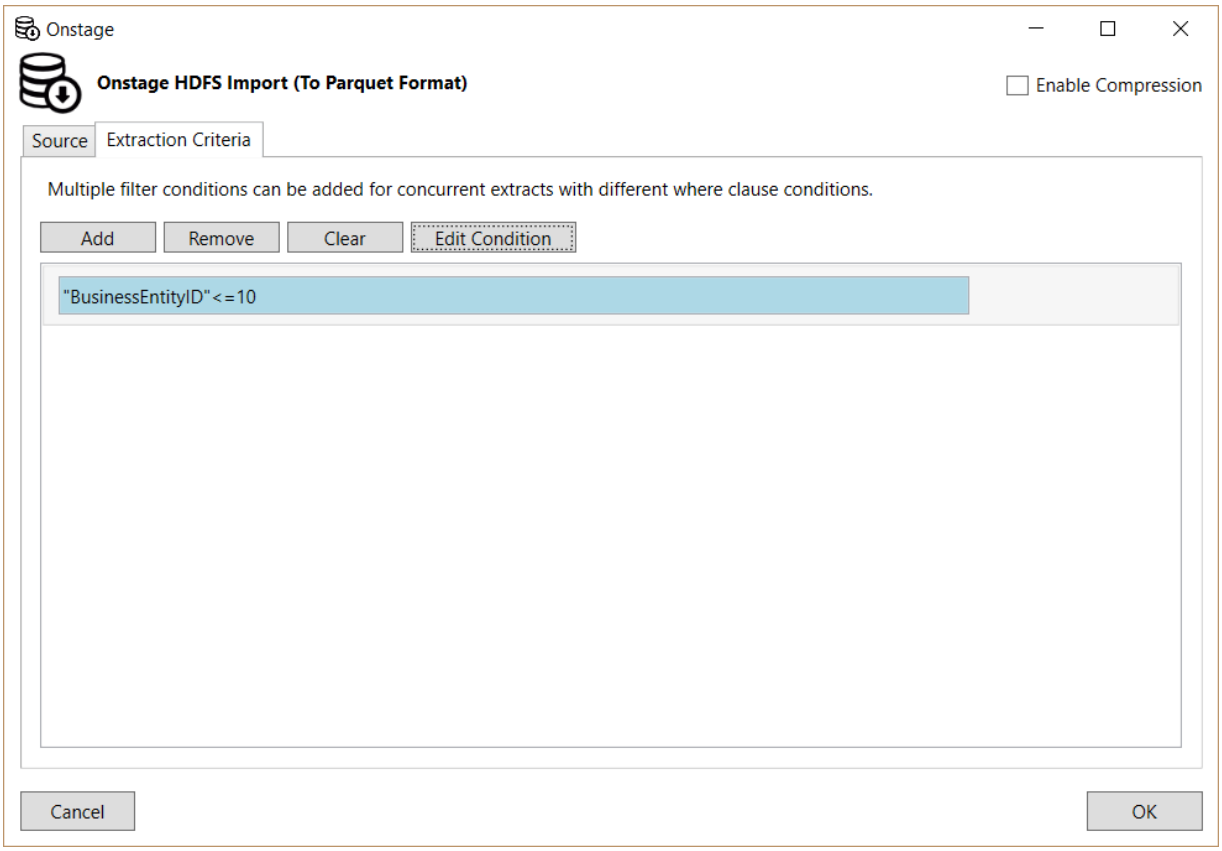
COLUMN		FUNCTION	
CONSTANT		UDF	
VARIABLE		SEQUENCE	
OPERATOR			

Expression

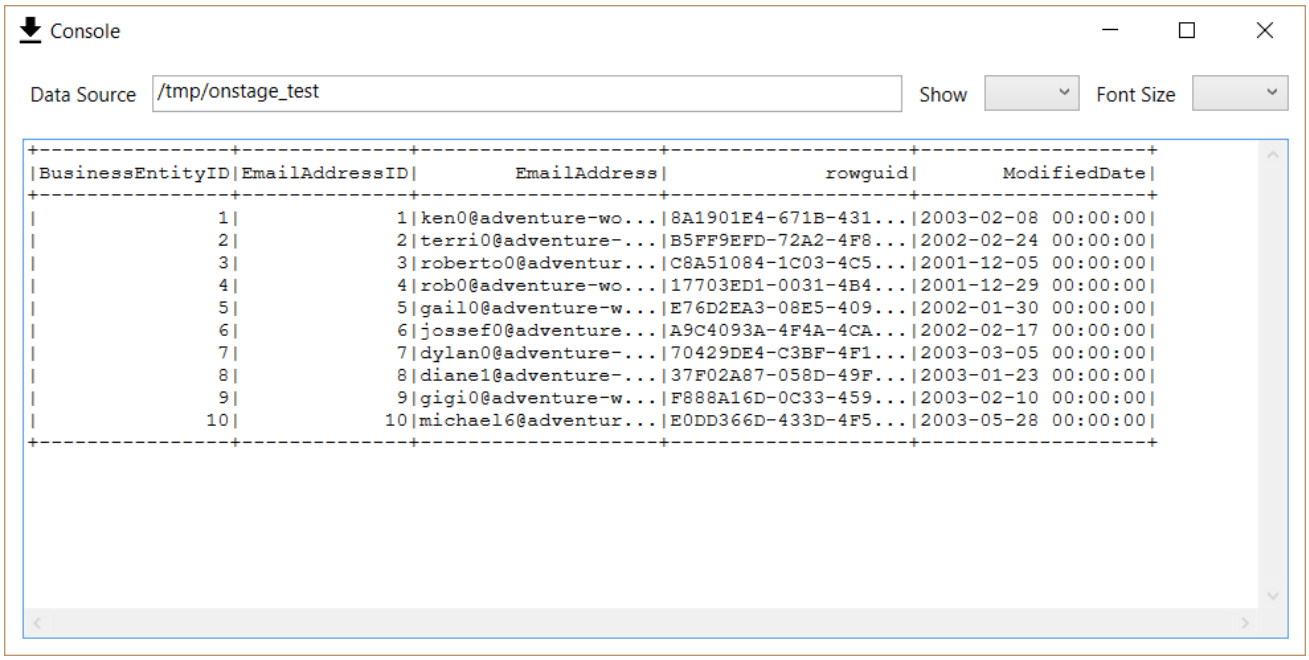
"BusinessEntityID"<=10

Cancel OK

Note that the syntax of the expression in the Extraction Criteria must reflect the SQL syntax of the source database where the extraction is taking place (in this example, SQL Server).



The results of the condition are not applied to the Preview inside the OnStage step. You can see the results of the condition by examining the output file:



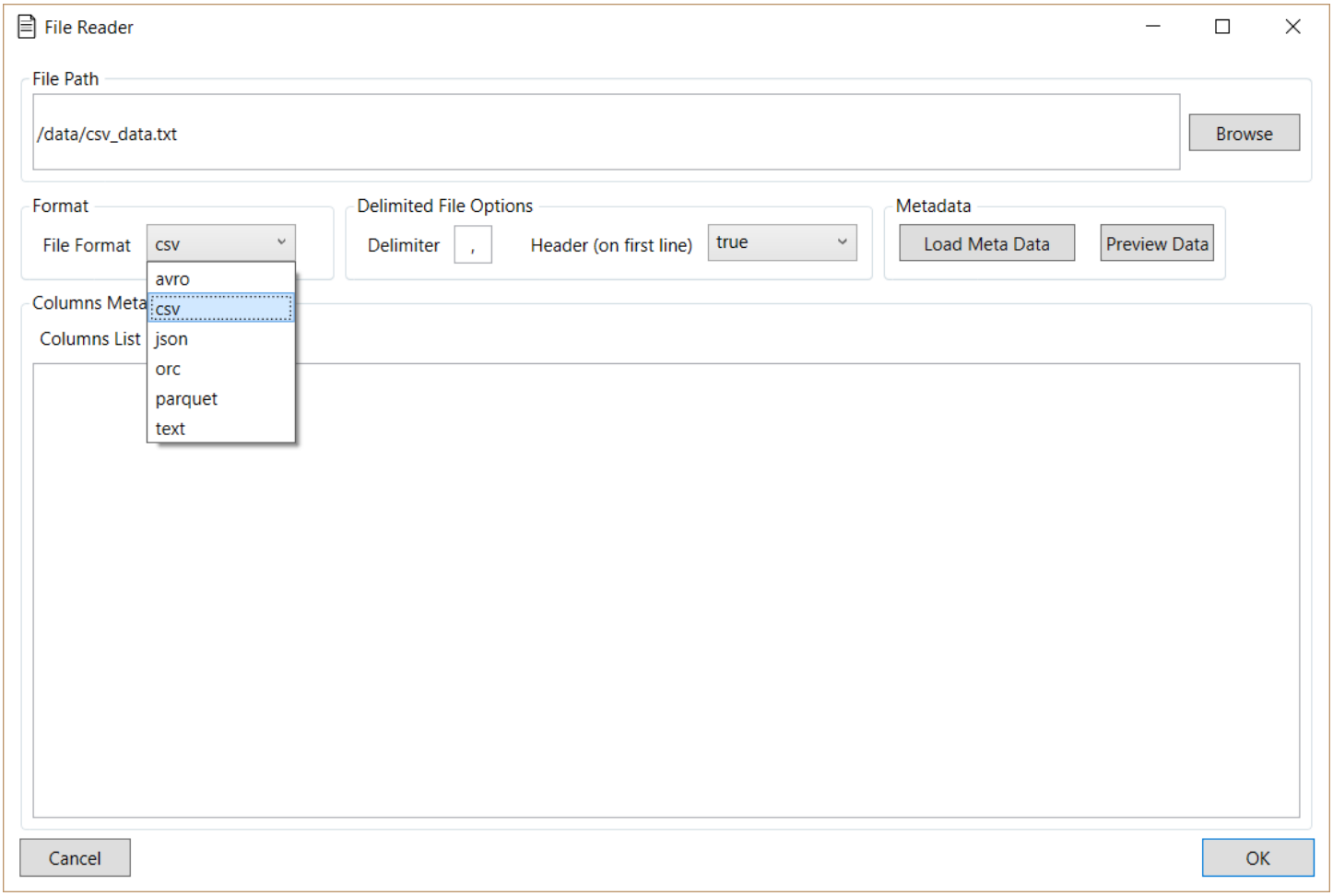

```
+----------------+--------------+--------------------+--------------------+-------------------+
|BusinessEntityID|EmailAddressID|        EmailAddress|             rowguid|       ModifiedDate|
+----------------+--------------+--------------------+--------------------+-------------------+
|               1|             1|ken0@adventure-wo...|8A1901E4-671B-431...|2003-02-08 00:00:00|
|               2|             2|terri0@adventure-...|B5FF9EFD-72A2-4F8...|2002-02-24 00:00:00|
|               3|             3|roberto0@adventur...|C8A51084-1C03-4C5...|2001-12-05 00:00:00|
|               4|             4|rob0@adventure-wo...|17703ED1-0031-4B4...|2001-12-29 00:00:00|
|               5|             5|gail0@adventure-w...|E76D2EA3-08E5-409...|2002-01-30 00:00:00|
|               6|             6|jossef0@adventure...|A9C4093A-4F4A-4CA...|2002-02-17 00:00:00|
|               7|             7|dylan0@adventure-...|70429DE4-C3BF-4F1...|2003-03-05 00:00:00|
|               8|             8|diane1@adventure-...|37F02A87-058D-49F...|2003-01-23 00:00:00|
|               9|             9|gigi0@adventure-w...|F888A16D-0C33-459...|2003-02-10 00:00:00|
|              10|            10|michael6@adventur...|E0DD366D-433D-4F5...|2003-05-28 00:00:00|
+----------------+--------------+--------------------+--------------------+-------------------+
```

File Reader

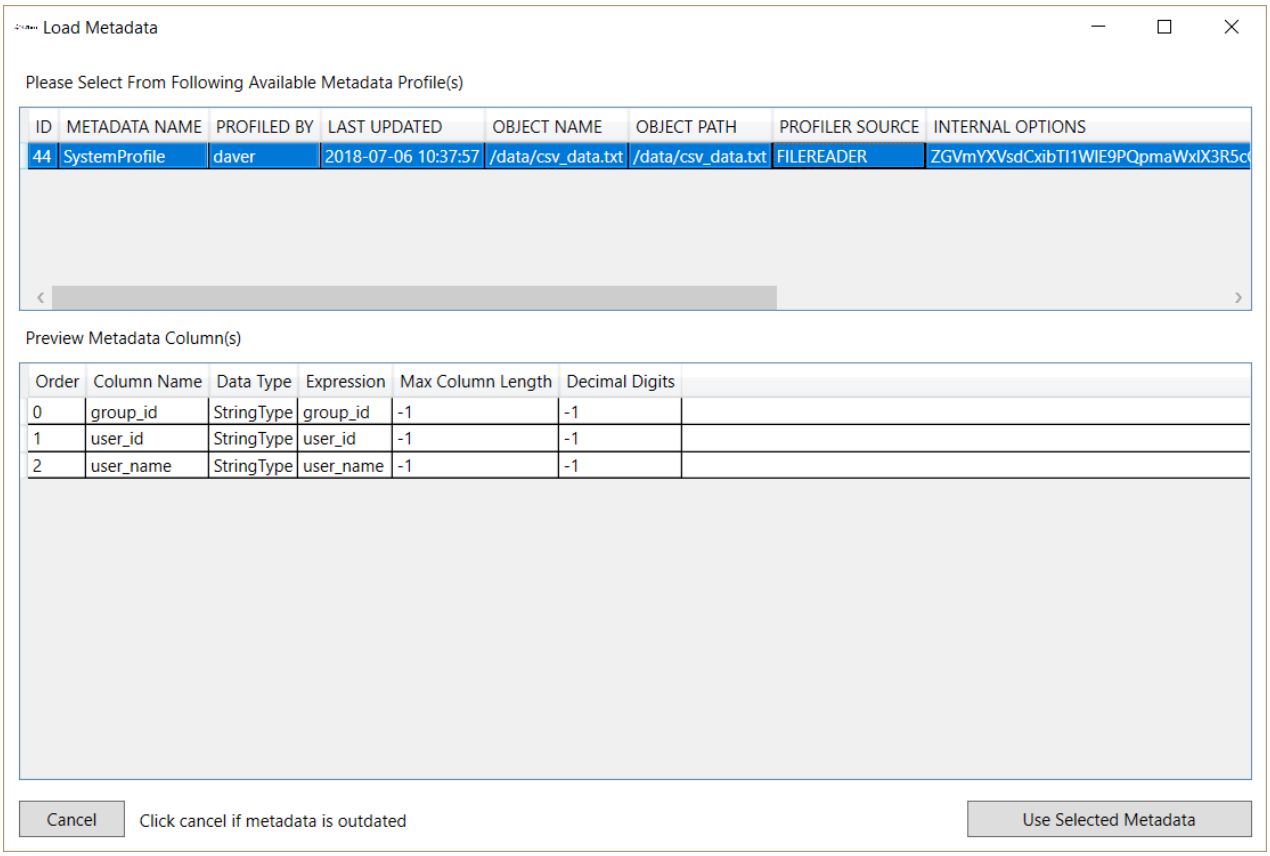
Introduction

File Reader step is used to load data from the HDFS file system.

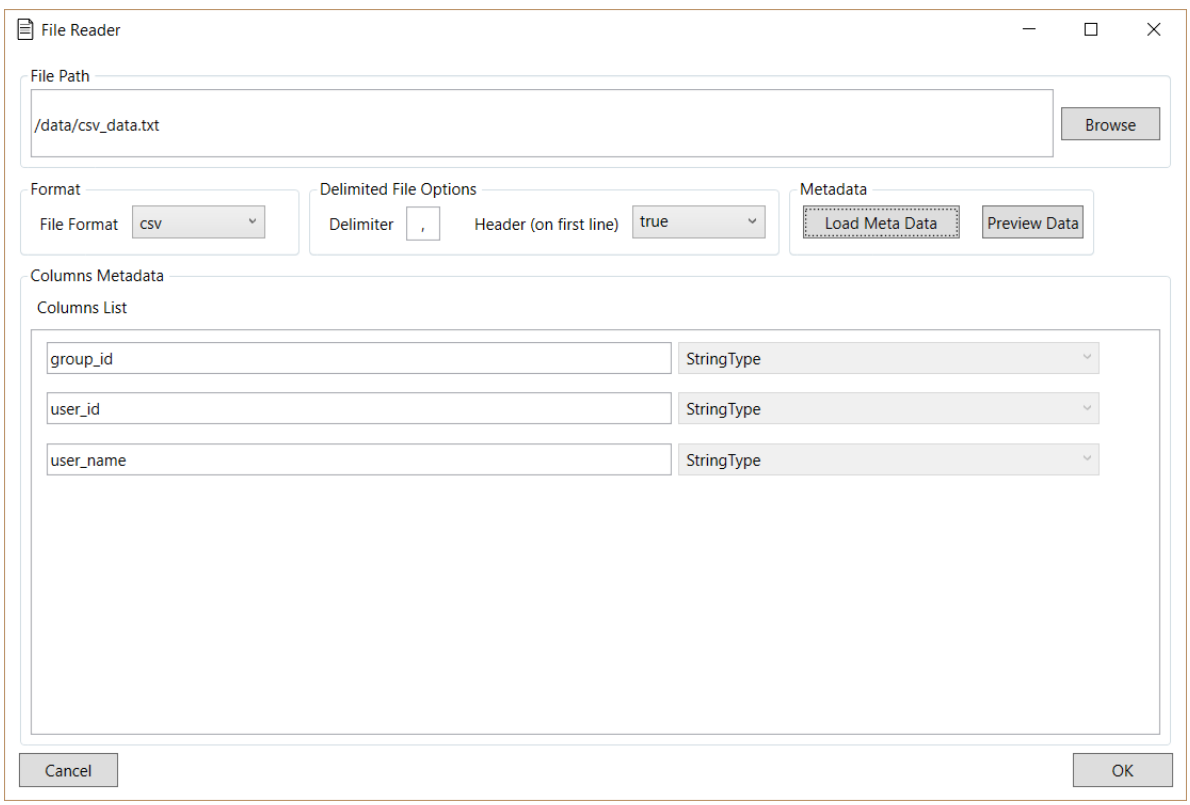
Example



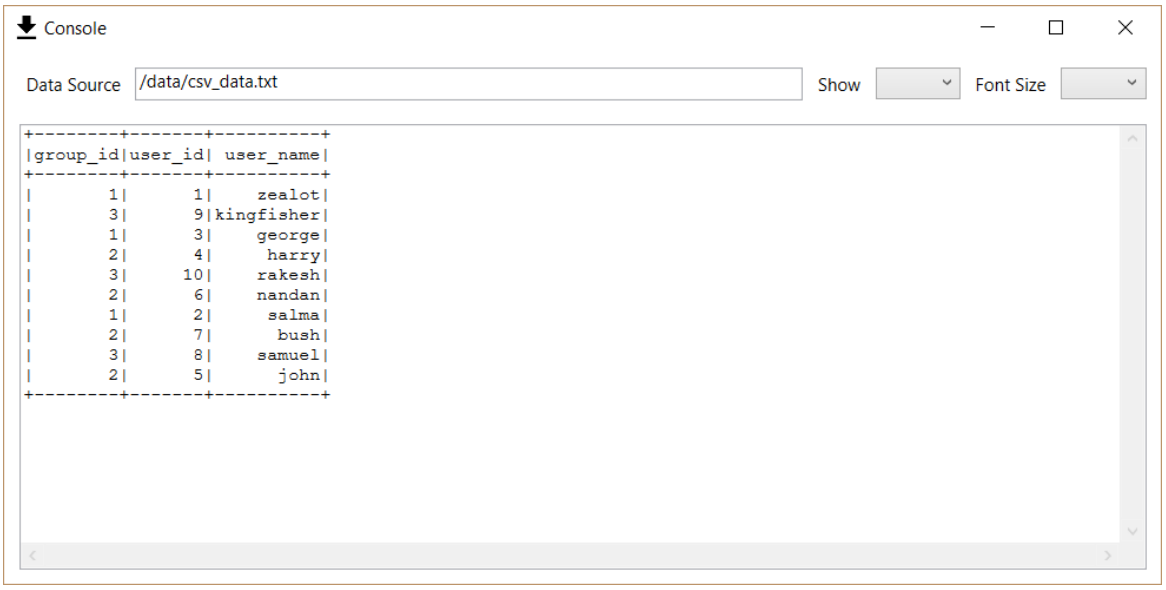
Clicking the Browse button will browse the Spark systems HDFS filesystem. After you have selected the file you wish to import, choose the file's format from the File Format dropdown list. In the example above, we are importing a CSV file named csv_data.txt. The Delimited File Options allow you to specify the delimiter and whether or not the first line contains a header. The Load Meta Data button checks to see of ELTMaestro has cached metadata associated with the file:



If the metadata is correct click Use Selected Metadata to have the metadata applied to the step otherwise click Cancel (in which case the metadata will be regenerated).



Use the Preview button to preview the data:

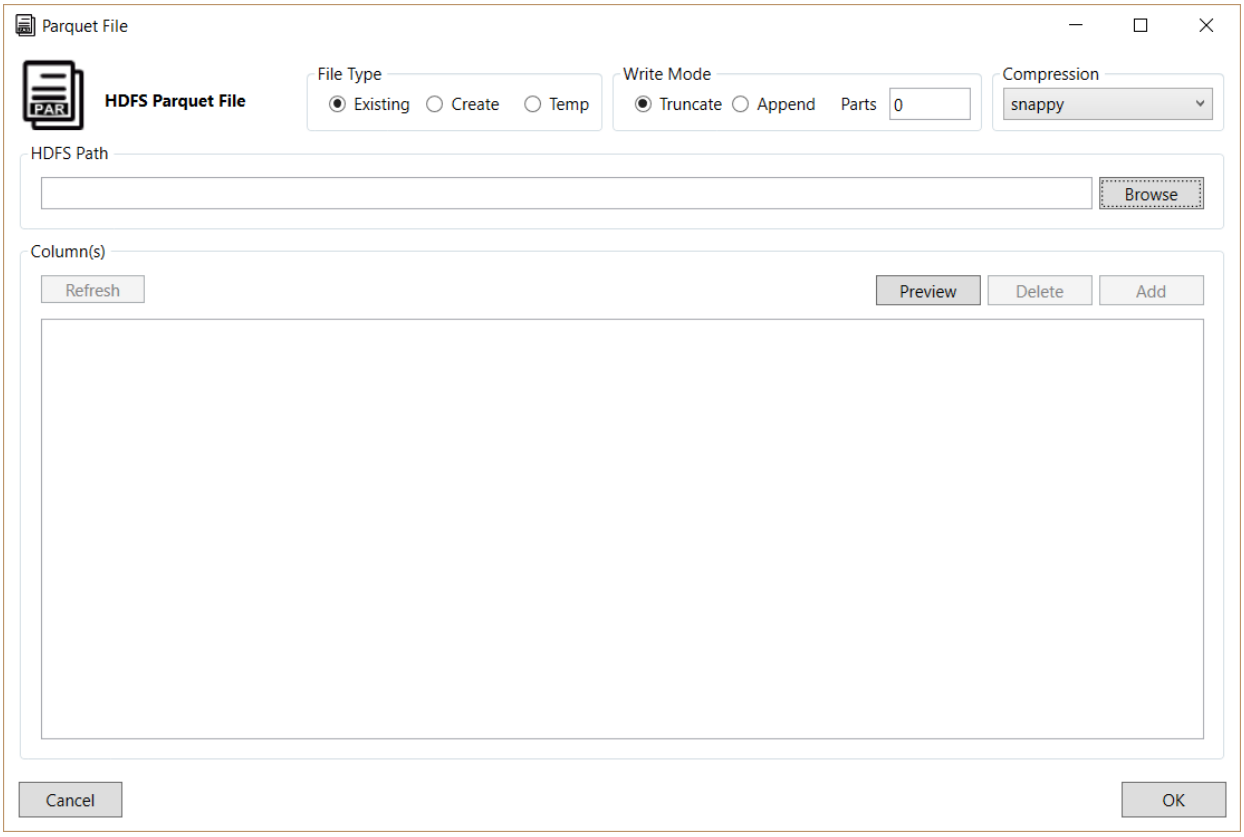


Parquet

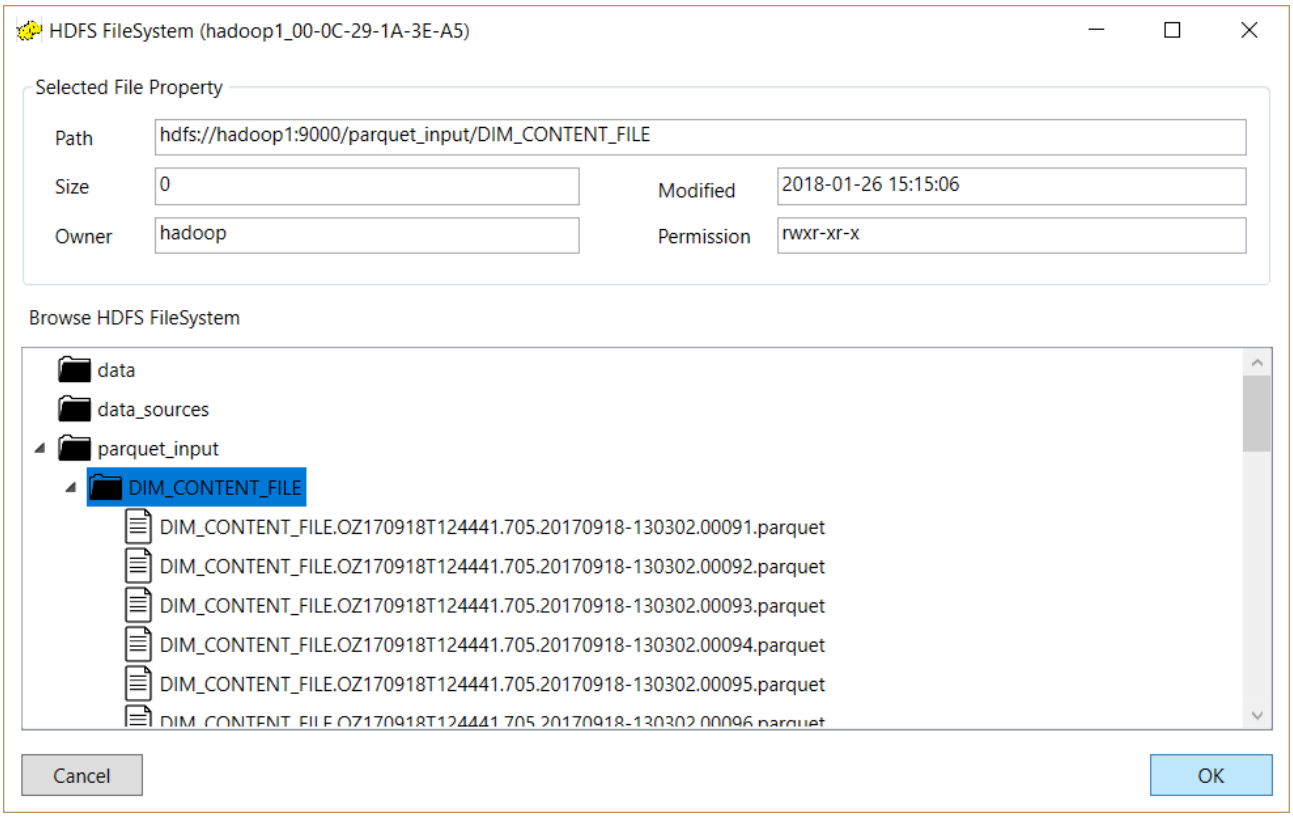
Introduction

Parquet step is used to read and write parquet files.

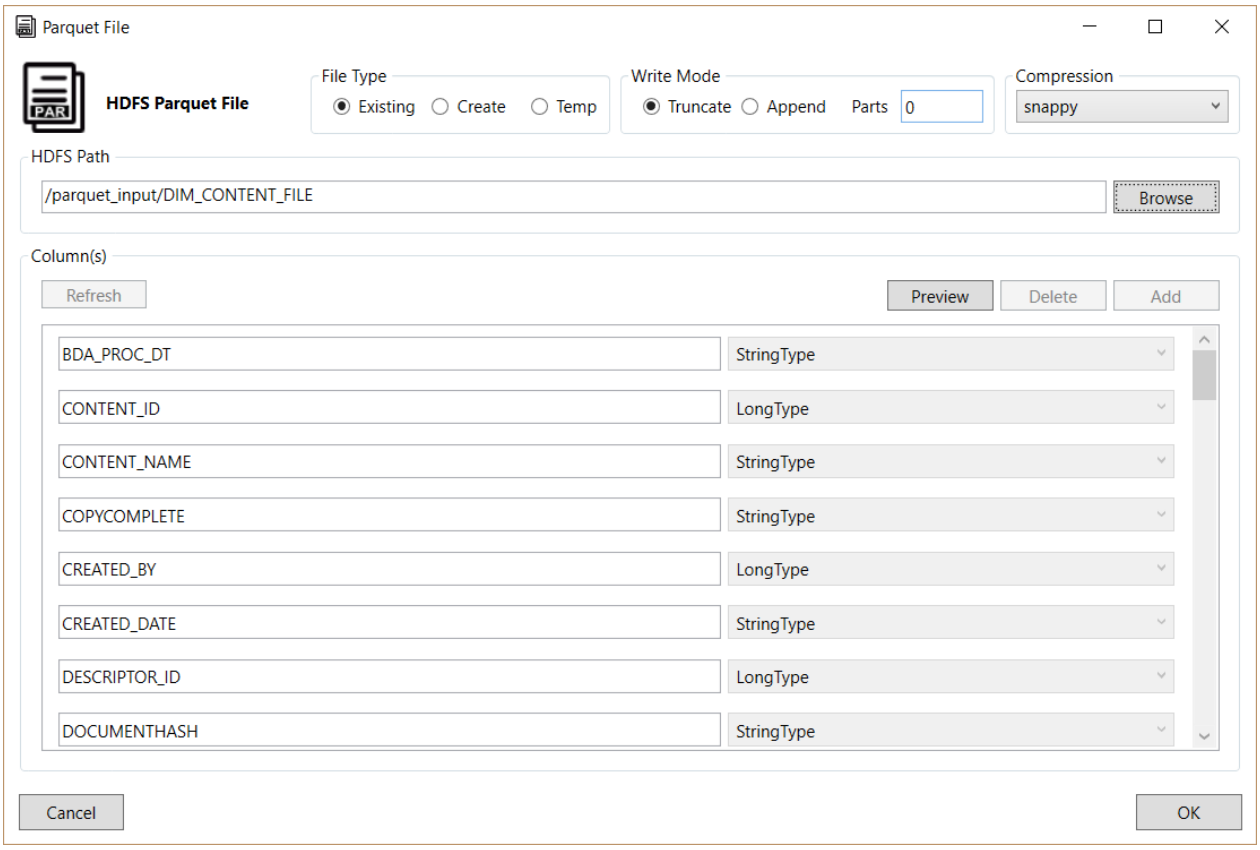
Example



To use the Parquet step for input, make sure the “Existing” File Type radio button is selected, then click on the Browse button to browse the HDFS filesystem:



Choose a Parquet file source, then click OK.



To use the Parquet step to write a Parquet file, choose File Type Create or Temp. You can specify whether to truncate or append to an existing file. You can specify a maximum number of parts for the Parquet file (0 means let the system decide). A dropdown list lets you choose among compression types snappy, gzip, and none.

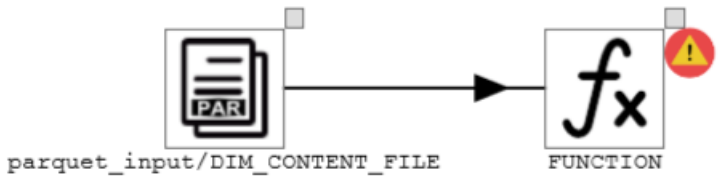
Function

Introduction

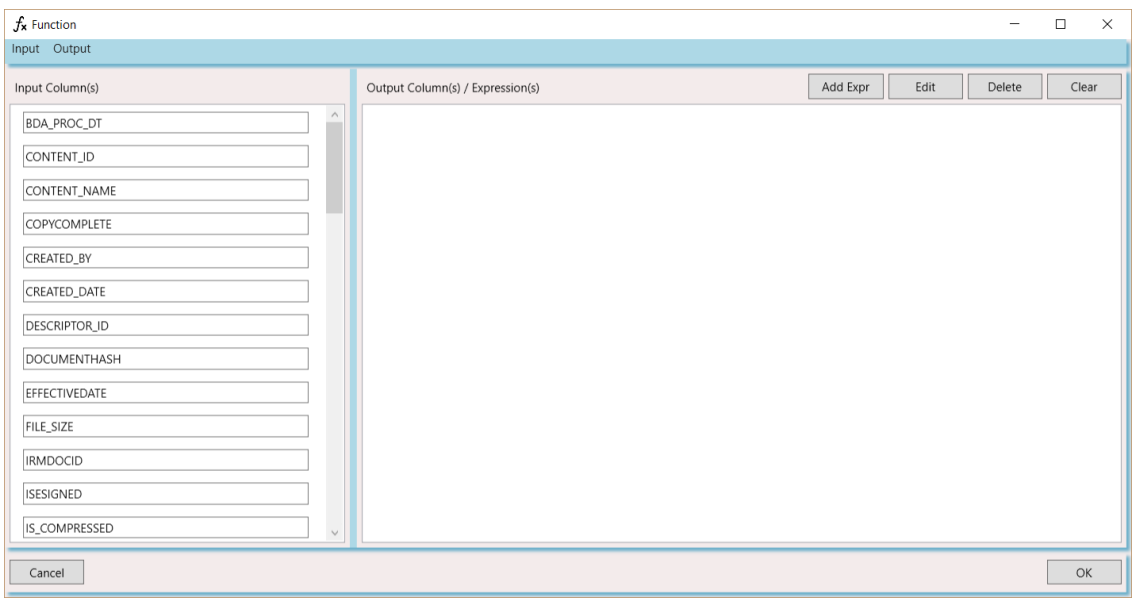
The Function step allows you to apply functions to columns and combinations of columns. The functions may be mathematical functions, string manipulation functions, date and time manipulation functions, conversion functions, or other functions appropriate to the data types. The Function step allows you to create new columns and to drop exiting columns.

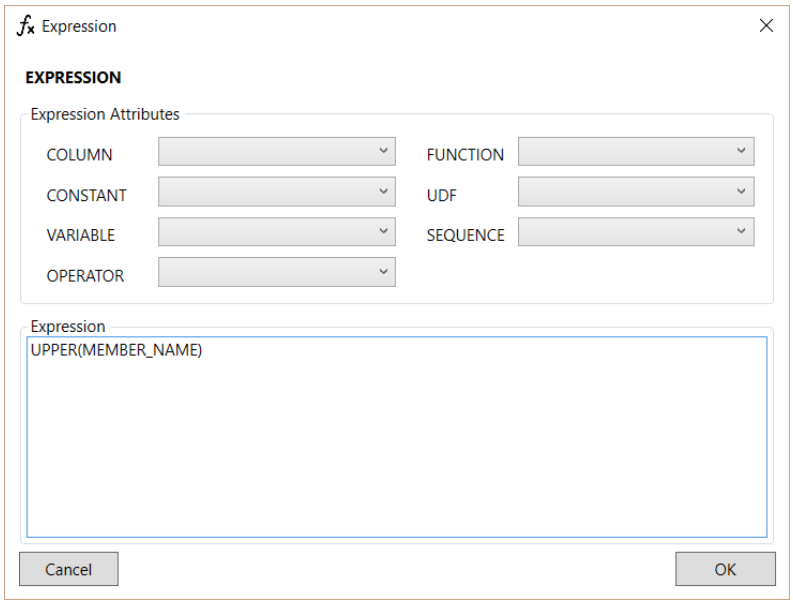
Example

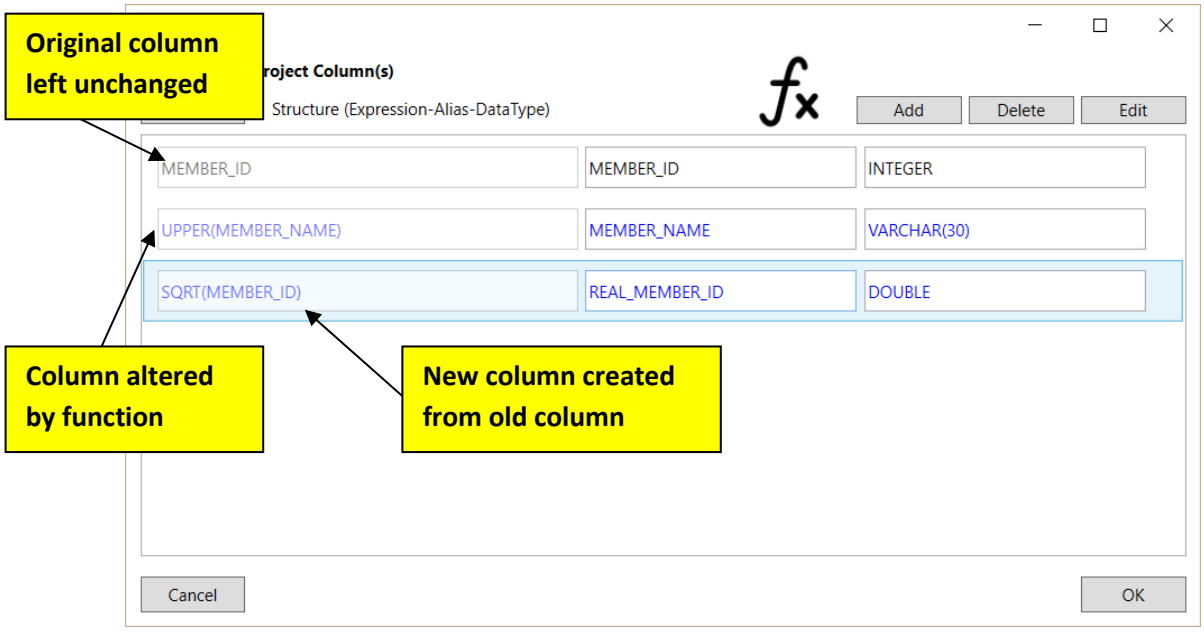
The Function step must be connected to a source of metadata for you to access its properties. Suppose the Function step is connected to a data source as shown below:



The properties window for the Function step will initially appear as follows:







Input and output for this example:

MEMBER_ID	MEMBER_NAME
1004	"dddd"
1002	"bbbb"
1001	"aaaa"
1003	"cccc"

MEMBER_ID	MEMBER_NAME	REAL_MEMBER_ID
1001	"AAAA"	31.63858403911275
1002	"BBBB"	31.654383582688826
1003	"CCCC"	31.670175244226233
1004	"DDDD"	31.68595903550972

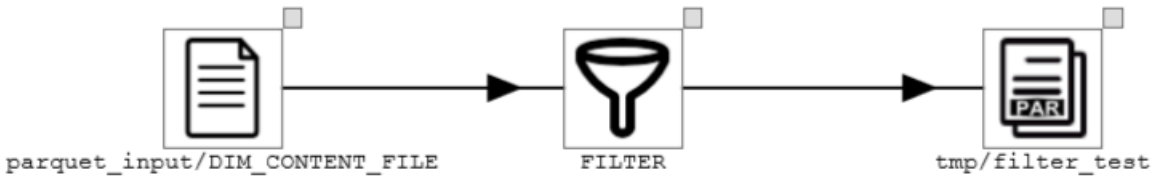
Filter

Introduction

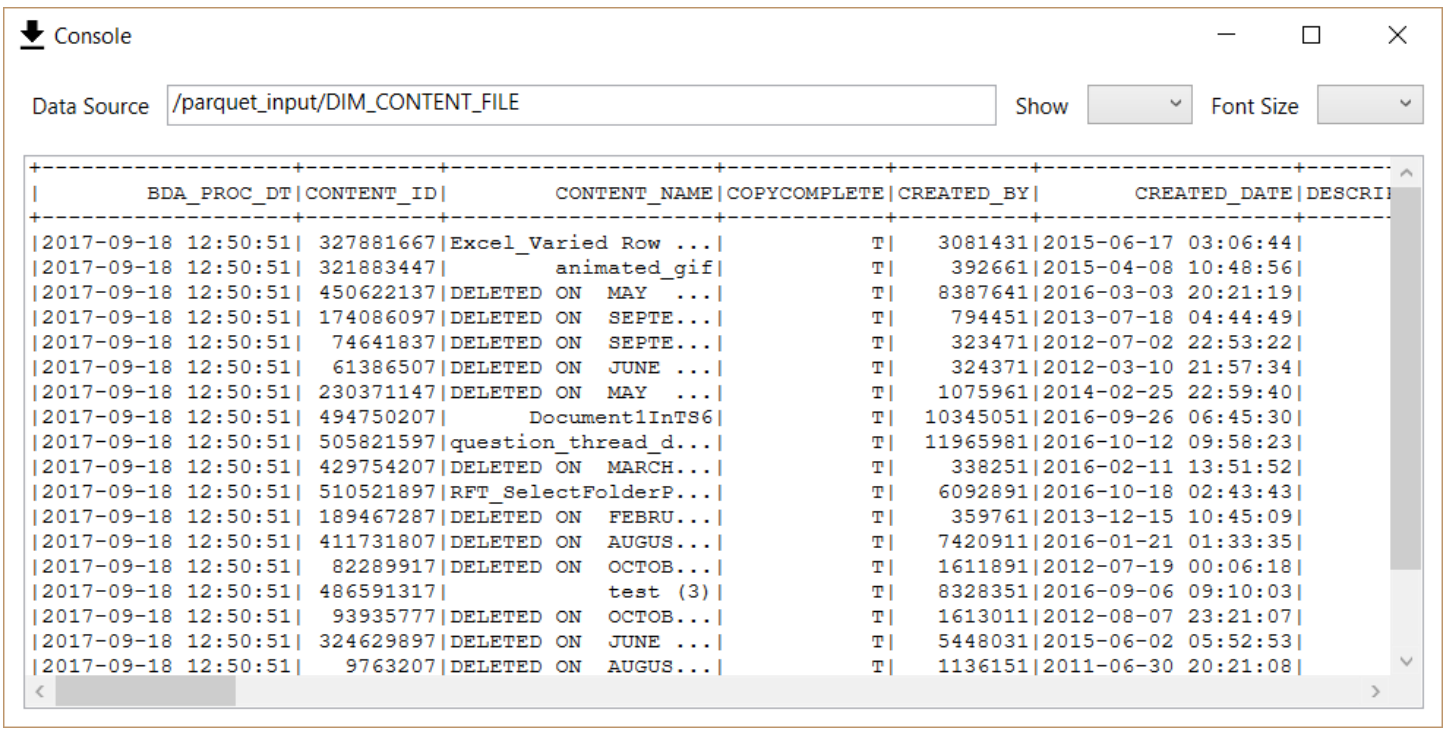
The Filter step allows you to filter the data flow, using an expression that, in SQL, would be placed in a WHERE clause. Note that the interface does not parse the expression for syntax errors before runtime; if the expression is complex or if you are unsure of your SQL syntax, it is best to try it out in an SQL parser beforehand.

Example

In this example, we use a Parquet file as input to the Filter step:

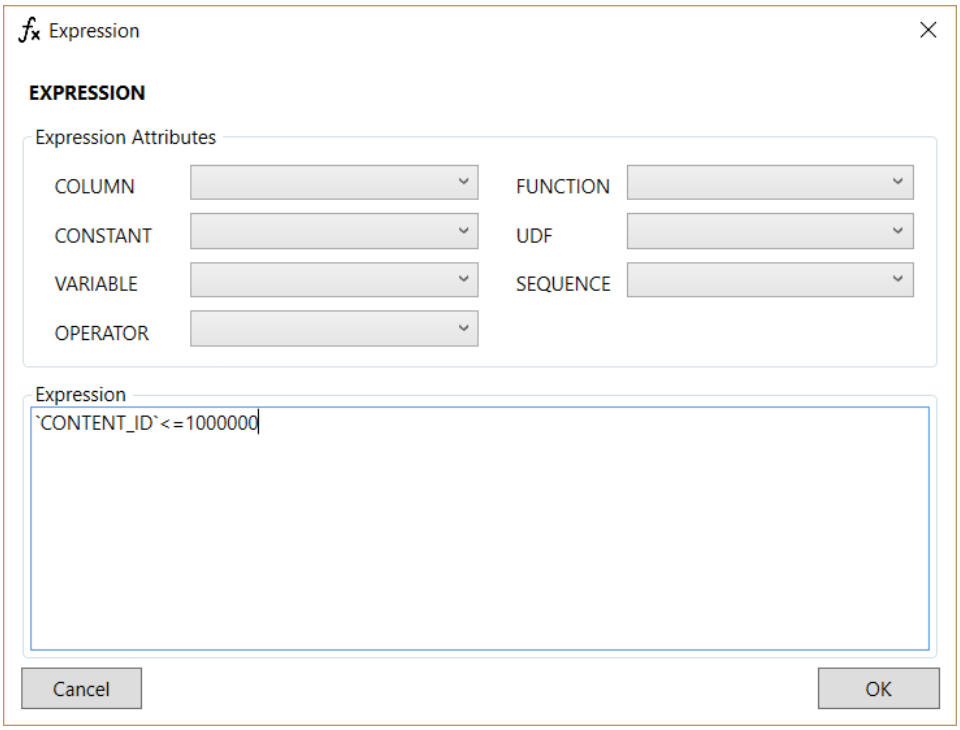


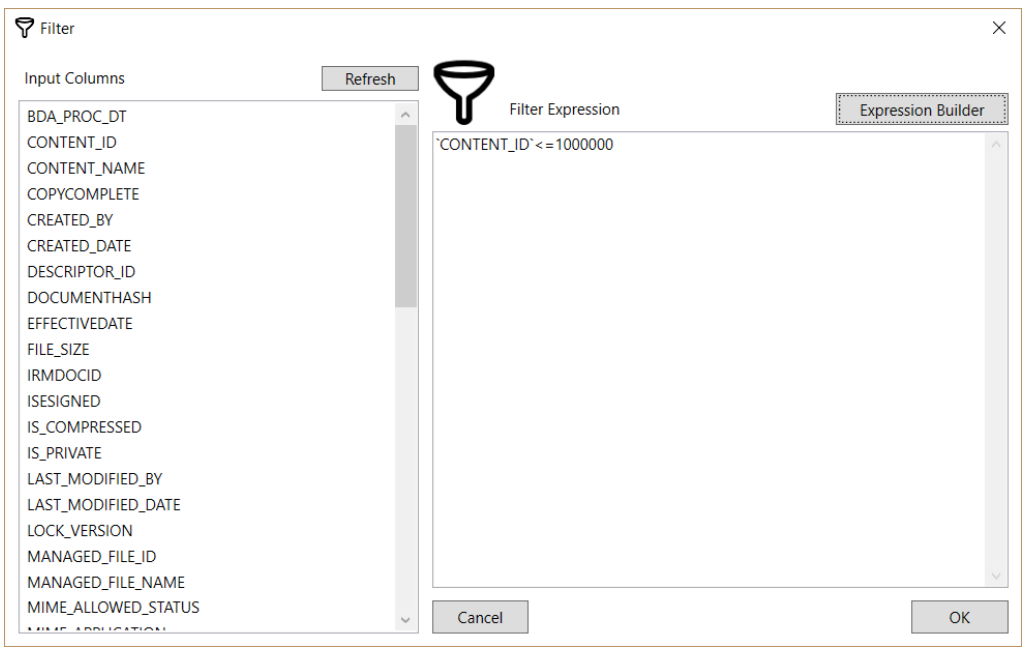
The input appears as follows:

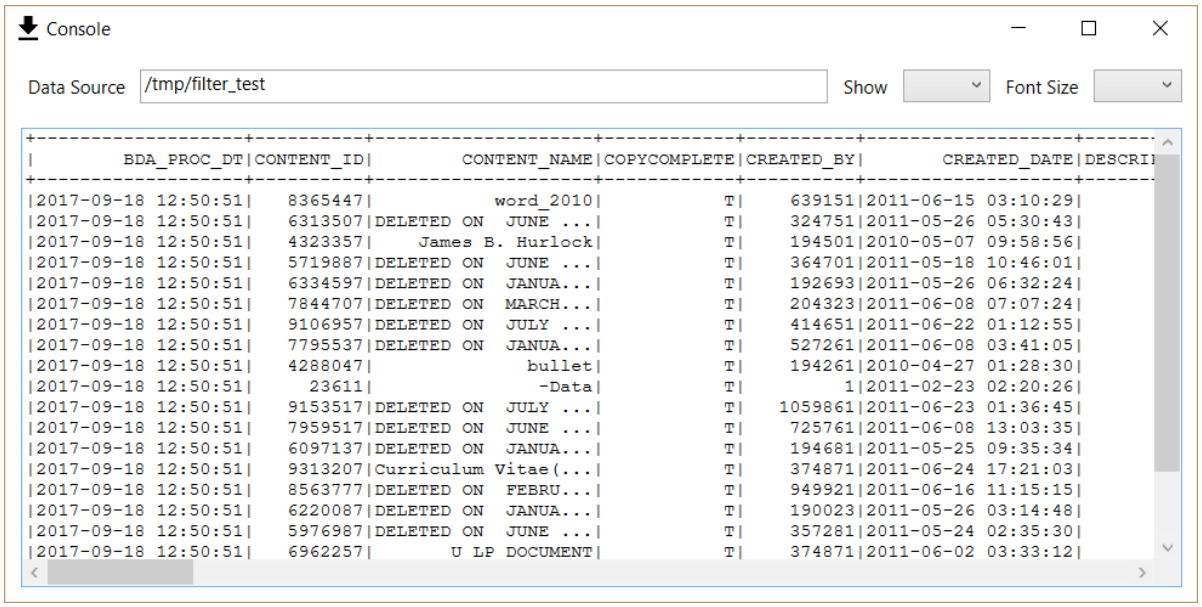


BDA_PROC_DT	CONTENT_ID	CONTENT_NAME	COPYCOMPLETE	CREATED_BY	CREATED_DATE	DESCRI
2017-09-18 12:50:51	327881667	Excel_Varied Row ...	T	3081431	2015-06-17 03:06:44	
2017-09-18 12:50:51	321883447	animated_gif	T	392661	2015-04-08 10:48:56	
2017-09-18 12:50:51	450622137	DELETED ON MAY ...	T	8387641	2016-03-03 20:21:19	
2017-09-18 12:50:51	174086097	DELETED ON SEPTE...	T	794451	2013-07-18 04:44:49	
2017-09-18 12:50:51	74641837	DELETED ON SEPTE...	T	323471	2012-07-02 22:53:22	
2017-09-18 12:50:51	61386507	DELETED ON JUNE ...	T	324371	2012-03-10 21:57:34	
2017-09-18 12:50:51	230371147	DELETED ON MAY ...	T	1075961	2014-02-25 22:59:40	
2017-09-18 12:50:51	494750207	Document1InT86	T	10345051	2016-09-26 06:45:30	
2017-09-18 12:50:51	505821597	question_thread_d...	T	11965981	2016-10-12 09:58:23	
2017-09-18 12:50:51	429754207	DELETED ON MARCH...	T	338251	2016-02-11 13:51:52	
2017-09-18 12:50:51	510521897	RFT_SelectFolderP...	T	6092891	2016-10-18 02:43:43	
2017-09-18 12:50:51	189467287	DELETED ON FEBRU...	T	359761	2013-12-15 10:45:09	
2017-09-18 12:50:51	411731807	DELETED ON AUGUS...	T	7420911	2016-01-21 01:33:35	
2017-09-18 12:50:51	82289917	DELETED ON OCTOB...	T	1611891	2012-07-19 00:06:18	
2017-09-18 12:50:51	486591317	test (3)	T	8328351	2016-09-06 09:10:03	
2017-09-18 12:50:51	93935777	DELETED ON OCTOB...	T	1613011	2012-08-07 23:21:07	
2017-09-18 12:50:51	324629897	DELETED ON JUNE ...	T	5448031	2015-06-02 05:52:53	
2017-09-18 12:50:51	9763207	DELETED ON AUGUS...	T	1136151	2011-06-30 20:21:08	

Inside the Filter step's properties, we click on Expression Builder, and enter an expression corresponding to a WHERE clause. Column names need to be surrounded by backquotes (`).


Expression
EXPRESSION
Expression Attributes
COLUMN
FUNCTION
CONSTANT
UDF
VARIABLE
SEQUENCE
OPERATOR
Expression
`CONTENT_ID`<=1000000
Cancel
OK


Filter
Input Columns
Refresh
BDA_PROC_DT
CONTENT_ID
CONTENT_NAME
COPYCOMPLETE
CREATED_BY
CREATED_DATE
DESCRIPTOR_ID
DOCUMENTHASH
EFFECTIVEDATE
FILE_SIZE
IRMDOCID
ISESIGNED
IS_COMPRESSED
IS_PRIVATE
LAST_MODIFIED_BY
LAST_MODIFIED_DATE
LOCK_VERSION
MANAGED_FILE_ID
MANAGED_FILE_NAME
MIME_ALLOWED_STATUS
Filter Expression
Expression Builder
`CONTENT_ID`<=1000000
Cancel
OK



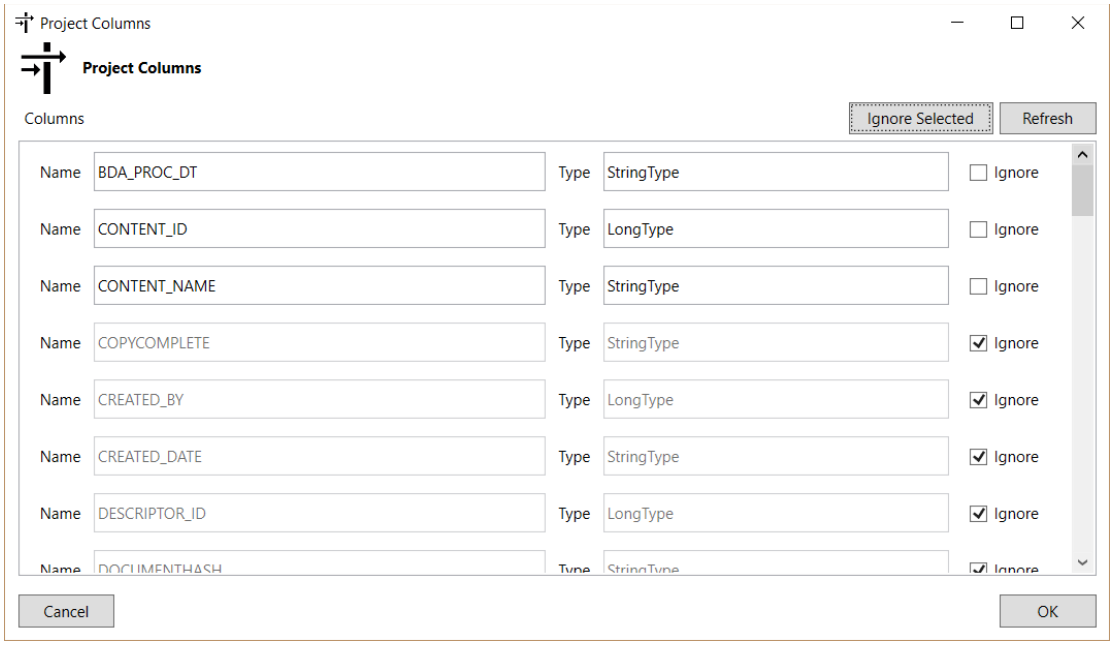
BDA_PROC_DT	CONTENT_ID	CONTENT_NAME	COPYCOMPLETE	CREATED_BY	CREATED_DATE	DESCRI
2017-09-18 12:50:51	8365447	word_2010	T	639151	2011-06-15 03:10:29	
2017-09-18 12:50:51	6313507	DELETED ON JUNE ...	T	324751	2011-05-26 05:30:43	
2017-09-18 12:50:51	4323357	James B. Hurlock	T	194501	2010-05-07 09:58:56	
2017-09-18 12:50:51	5719887	DELETED ON JUNE ...	T	364701	2011-05-18 10:46:01	
2017-09-18 12:50:51	6334597	DELETED ON JANUA...	T	192693	2011-05-26 06:32:24	
2017-09-18 12:50:51	7844707	DELETED ON MARCH...	T	204323	2011-06-08 07:07:24	
2017-09-18 12:50:51	9106957	DELETED ON JULY ...	T	414651	2011-06-22 01:12:55	
2017-09-18 12:50:51	7795537	DELETED ON JANUA...	T	527261	2011-06-08 03:41:05	
2017-09-18 12:50:51	4288047	bullet	T	194261	2010-04-27 01:28:30	
2017-09-18 12:50:51	23611	-Data	T	1	2011-02-23 02:20:26	
2017-09-18 12:50:51	9153517	DELETED ON JULY ...	T	1059861	2011-06-23 01:36:45	
2017-09-18 12:50:51	7959517	DELETED ON JUNE ...	T	725761	2011-06-08 13:03:35	
2017-09-18 12:50:51	6097137	DELETED ON JANUA...	T	194681	2011-05-25 09:35:34	
2017-09-18 12:50:51	9313207	Curriculum Vitae(...	T	374871	2011-06-24 17:21:03	
2017-09-18 12:50:51	8563777	DELETED ON FEBRU...	T	949921	2011-06-16 11:15:15	
2017-09-18 12:50:51	6220087	DELETED ON JANUA...	T	190023	2011-05-26 03:14:48	
2017-09-18 12:50:51	5976987	DELETED ON JUNE ...	T	357281	2011-05-24 02:35:30	
2017-09-18 12:50:51	6962257	U LP DOCUMENT	T	374871	2011-06-02 03:33:12	

The output only contains rows where CONTENT_ID <= 10000000.

Projection

Introduction

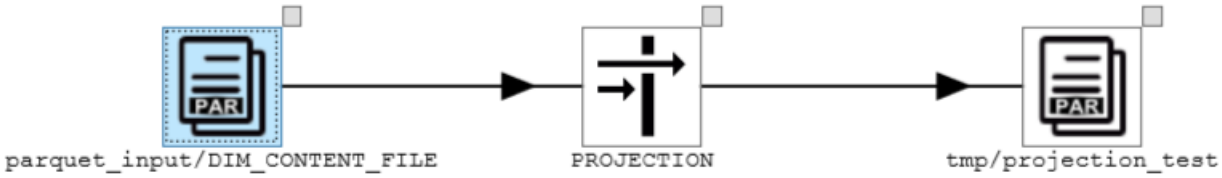
The Project Columns step allows you to drop some of the columns from a dataset. It has a very straightforward interface.



Example,

Simply check Ignore on the columns you wish to drop.

The example above is taken from the following job:



The input and output are shown below:

Console

Data Source /parquet_input/DIM_CONTENT_FILE Show Font Size

```
+-------------------+----------+--------------------+------------+----------+-------------------+------
|        BDA_PROC_DT|CONTENT_ID|        CONTENT_NAME|COPYCOMPLETE|CREATED_BY|       CREATED_DATE|DESCRI
+-------------------+----------+--------------------+------------+----------+-------------------+------
|2017-09-18 12:50:51| 327881667|Excel_Varied Row ...|           T|   3081431|2015-06-17 03:06:44|
|2017-09-18 12:50:51| 321883447|        animated_gif|           T|    392661|2015-04-08 10:48:56|
|2017-09-18 12:50:51| 450622137|DELETED ON  MAY  ...|           T|   8387641|2016-03-03 20:21:19|
|2017-09-18 12:50:51| 174086097|DELETED ON  SEPTE...|           T|    794451|2013-07-18 04:44:49|
|2017-09-18 12:50:51|  74641837|DELETED ON  SEPTE...|           T|    323471|2012-07-02 22:53:22|
|2017-09-18 12:50:51|  61386507|DELETED ON  JUNE ...|           T|    324371|2012-03-10 21:57:34|
|2017-09-18 12:50:51| 230371147|DELETED ON  MAY  ...|           T|   1075961|2014-02-25 22:59:40|
|2017-09-18 12:50:51| 494750207|      Document1InTS6|           T|  10345051|2016-09-26 06:45:30|
|2017-09-18 12:50:51| 505821597|question_thread_d...|           T|  11965981|2016-10-12 09:58:23|
|2017-09-18 12:50:51| 429754207|DELETED ON  MARCH...|           T|    338251|2016-02-11 13:51:52|
|2017-09-18 12:50:51| 510521897|RFT_SelectFolderP...|           T|   6092891|2016-10-18 02:43:43|
|2017-09-18 12:50:51| 189467287|DELETED ON  FEBRU...|           T|    359761|2013-12-15 10:45:09|
|2017-09-18 12:50:51| 411731807|DELETED ON  AUGUS...|           T|   7420911|2016-01-21 01:33:35|
|2017-09-18 12:50:51|  82289917|DELETED ON  OCTOB...|           T|   1611891|2012-07-19 00:06:18|
|2017-09-18 12:50:51| 486591317|            test (3)|           T|   8328351|2016-09-06 09:10:03|
|2017-09-18 12:50:51|  93935777|DELETED ON  OCTOB...|           T|   1613011|2012-08-07 23:21:07|
|2017-09-18 12:50:51| 324629897|DELETED ON  JUNE ...|           T|   5448031|2015-06-02 05:52:53|
|2017-09-18 12:50:51|   9763207|DELETED ON  AUGUS...|           T|   1136151|2011-06-30 20:21:08|
```

Console

Data Source /tmp/projection_test Show Font Size

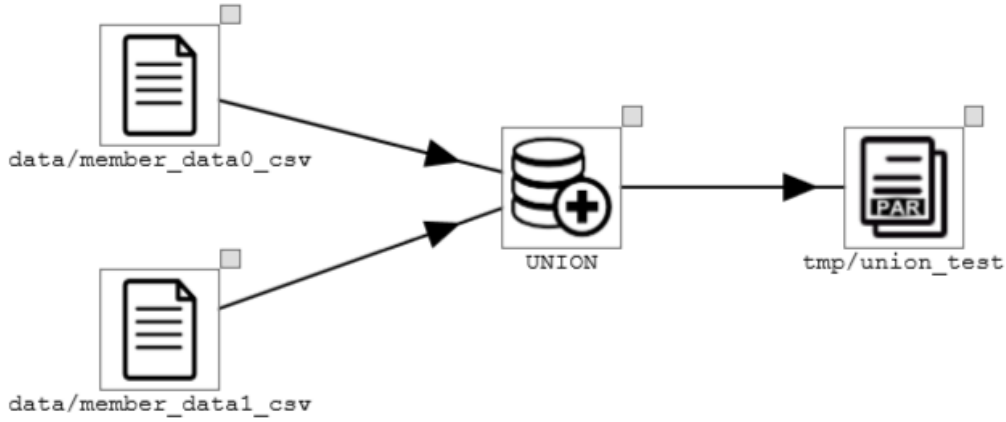
```
+-------------------+----------+--------------------+
|        BDA_PROC_DT|CONTENT_ID|        CONTENT_NAME|
+-------------------+----------+--------------------+
|2017-09-18 12:50:51| 327881667|Excel_Varied Row ...|
|2017-09-18 12:50:51| 321883447|        animated_gif|
|2017-09-18 12:50:51| 450622137|DELETED ON  MAY  ...|
|2017-09-18 12:50:51| 174086097|DELETED ON  SEPTE...|
|2017-09-18 12:50:51|  74641837|DELETED ON  SEPTE...|
|2017-09-18 12:50:51|  61386507|DELETED ON  JUNE ...|
|2017-09-18 12:50:51| 230371147|DELETED ON  MAY  ...|
|2017-09-18 12:50:51| 494750207|      Document1InTS6|
|2017-09-18 12:50:51| 505821597|question_thread_d...|
|2017-09-18 12:50:51| 429754207|DELETED ON  MARCH...|
|2017-09-18 12:50:51| 510521897|RFT_SelectFolderP...|
|2017-09-18 12:50:51| 189467287|DELETED ON  FEBRU...|
|2017-09-18 12:50:51| 411731807|DELETED ON  AUGUS...|
|2017-09-18 12:50:51|  82289917|DELETED ON  OCTOB...|
|2017-09-18 12:50:51| 486591317|            test (3)|
|2017-09-18 12:50:51|  93935777|DELETED ON  OCTOB...|
|2017-09-18 12:50:51| 324629897|DELETED ON  JUNE ...|
|2017-09-18 12:50:51|   9763207|DELETED ON  AUGUS...|
```

Union

Introduction

The Union step combines two datasets. The datasets have to have the same column metadata.

Example,



Here, the Union step is used to combine the contents of two flat files, and the result is loaded to a table.

The contents of the first flat file is

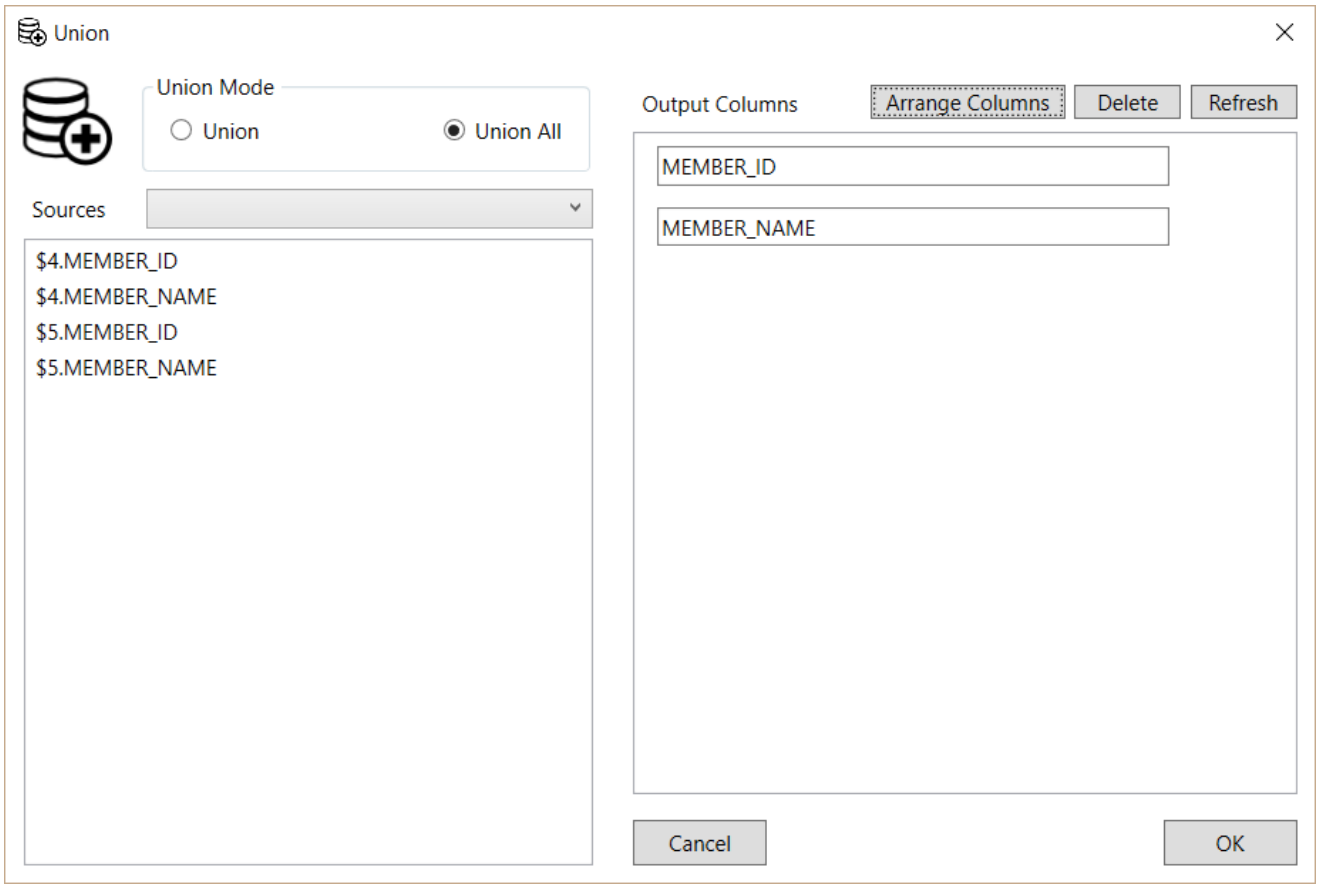
```
+---------+-----------+
|MEMBER_ID|MEMBER_NAME|
+---------+-----------+
|     1001|       aaaa|
|     1002|       bbbb|
|     1003|       cccc|
|     1004|       dddd|
|     1005|       eeee|
|     1006|       ffff|
+---------+-----------+
```

and the contents of the second flat file is

.

```
+---------+-----------+
|MEMBER_ID|MEMBER_NAME|
+---------+-----------+
|     1001|       aaaa|
|     1002|       bbbb|
|     1003|       cccc|
|     1004|       dddd|
|     1021|       uuuu|
|     1022|       vvvv|
+---------+-----------+
```

The Union step properties window is set up as follows:



In the Sources drop-down list choose ALL.

The output is:

```
+---------+-----------+
|MEMBER_ID|MEMBER_NAME|
+---------+-----------+
|     1001|       aaaa|
|     1002|       bbbb|
|     1003|       cccc|
|     1004|       dddd|
|     1021|       uuuu|
|     1022|       vvvv|
|     1001|       aaaa|
|     1002|       bbbb|
|     1003|       cccc|
|     1004|       dddd|
|     1005|       eeee|
|     1006|       ffff|
+---------+-----------+
```

Only the Union All option is available in ELTMaestro for Spark. (In other ELTMaestro editions, the Union option would remove duplicates.)

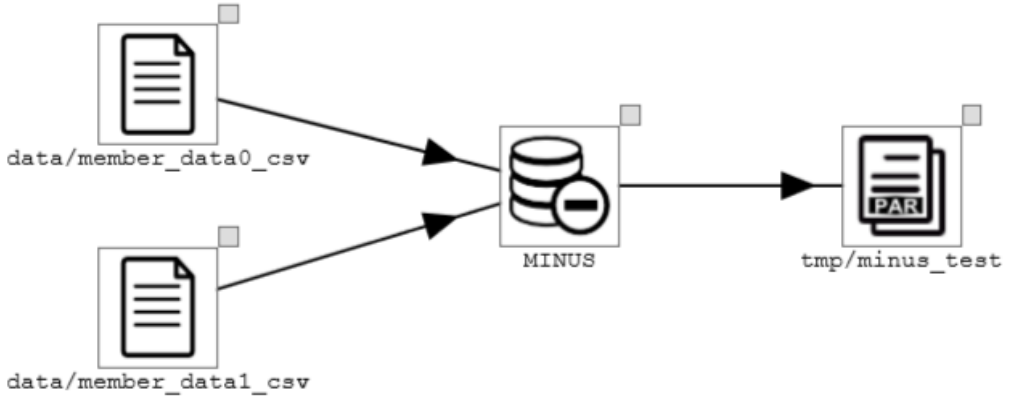
Minus

Introduction

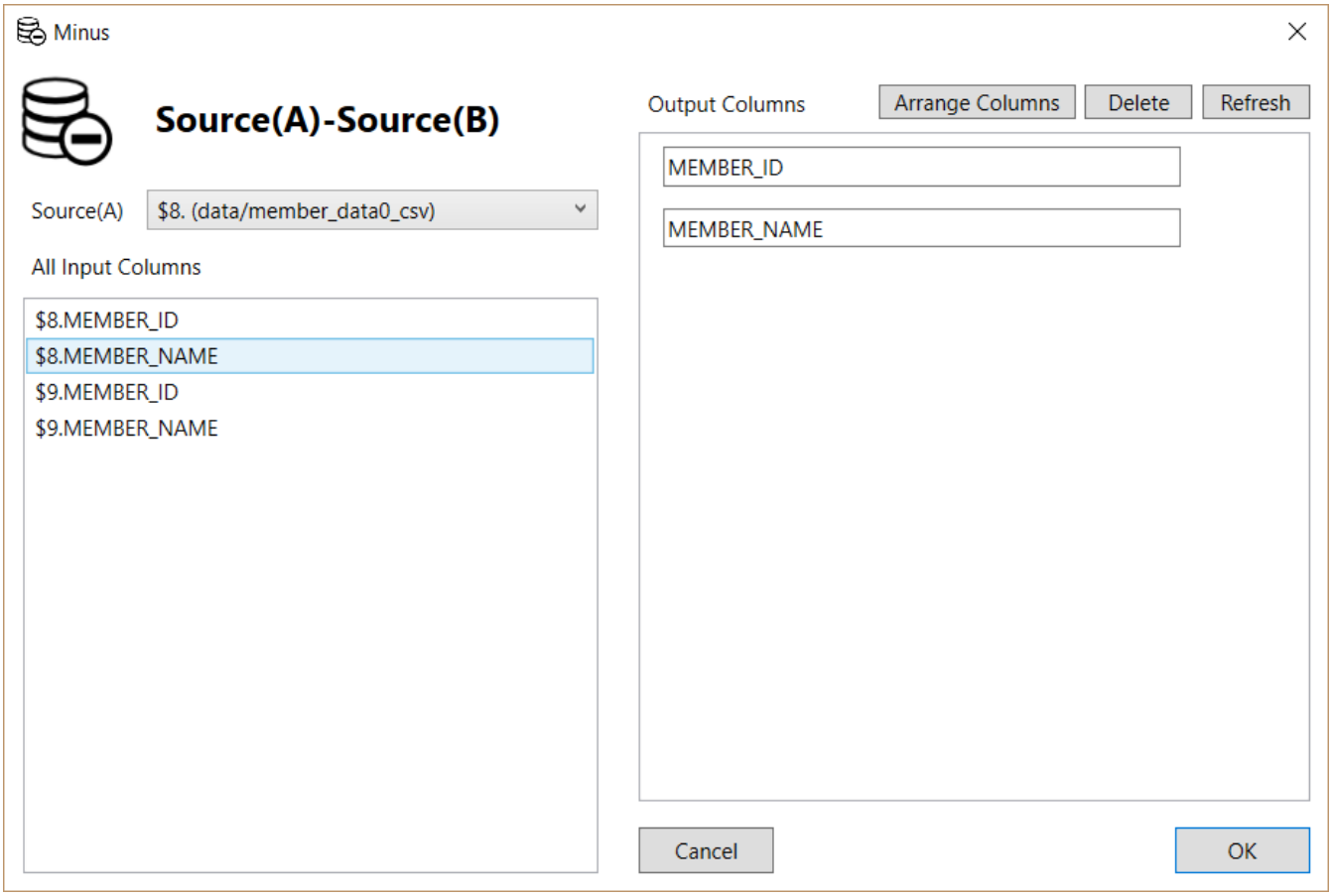
The Minus step subtracts the contents of one dataset from another.

Example

Consider the job shown below:



In this job, the contents of the table MEMBER_NAMES_2 are subtracted from the contents of the table MEMBER_NAMES. The column metadata for both inputs must be the same. In the properties window, you specify which dataset is the minuend (i.e. Source(A)) and which dataset is the subtrahend (i.e. Source(B)).



All rows in Source(A) matching any row in Source(B) will be removed from the result, regardless of how many duplicates there are. For example if the inputs are as follows:

Source(A):

MEMBER_ID	MEMBER_NAME
1001	aaaa
1002	bbbb
1003	cccc
1004	dddd
1005	eeee
1006	ffff

Source(B):

MEMBER_ID	MEMBER_NAME
1001	aaaa
1002	bbbb
1003	cccc
1004	dddd
1021	uuuu
1022	vvvv

The output will be:

MEMBER_ID	MEMBER_NAME
1005	eeee
1006	ffff

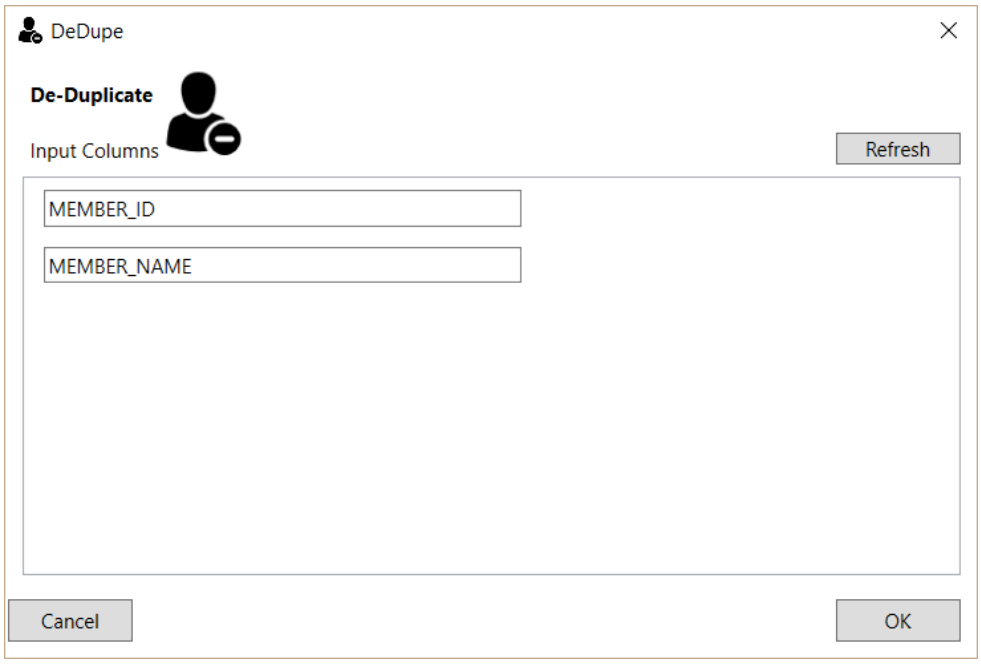
Dedupe

Introduction

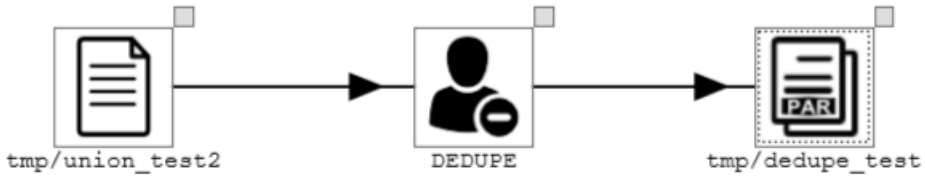
The Dedupe step removes duplicate rows from a dataset.

Example

There are no properties to set.



Consider the following job:



Input	Output (after dedupe)
+---------+----------+ |MEMBER_ID|MEMBER_NAME| +---------+----------+ | 1001| aaaa| | 1002| bbbb| | 1003| cccc| | 1004| dddd| | 1021| uuuu| | 1022| vvvv| | 1001| aaaa| | 1002| bbbb| | 1003| cccc| | 1004| dddd| | 1021| uuuu| | 1022| vvvv| +---------+----------+	+---------+----------+ |MEMBER_ID|MEMBER_NAME| +---------+----------+ | 1002| bbbb| | 1021| uuuu| | 1022| vvvv| | 1004| dddd| | 1001| aaaa| | 1003| cccc| +---------+----------+

Σ Aggregate

Introduction

The Aggregate step performs aggregations.

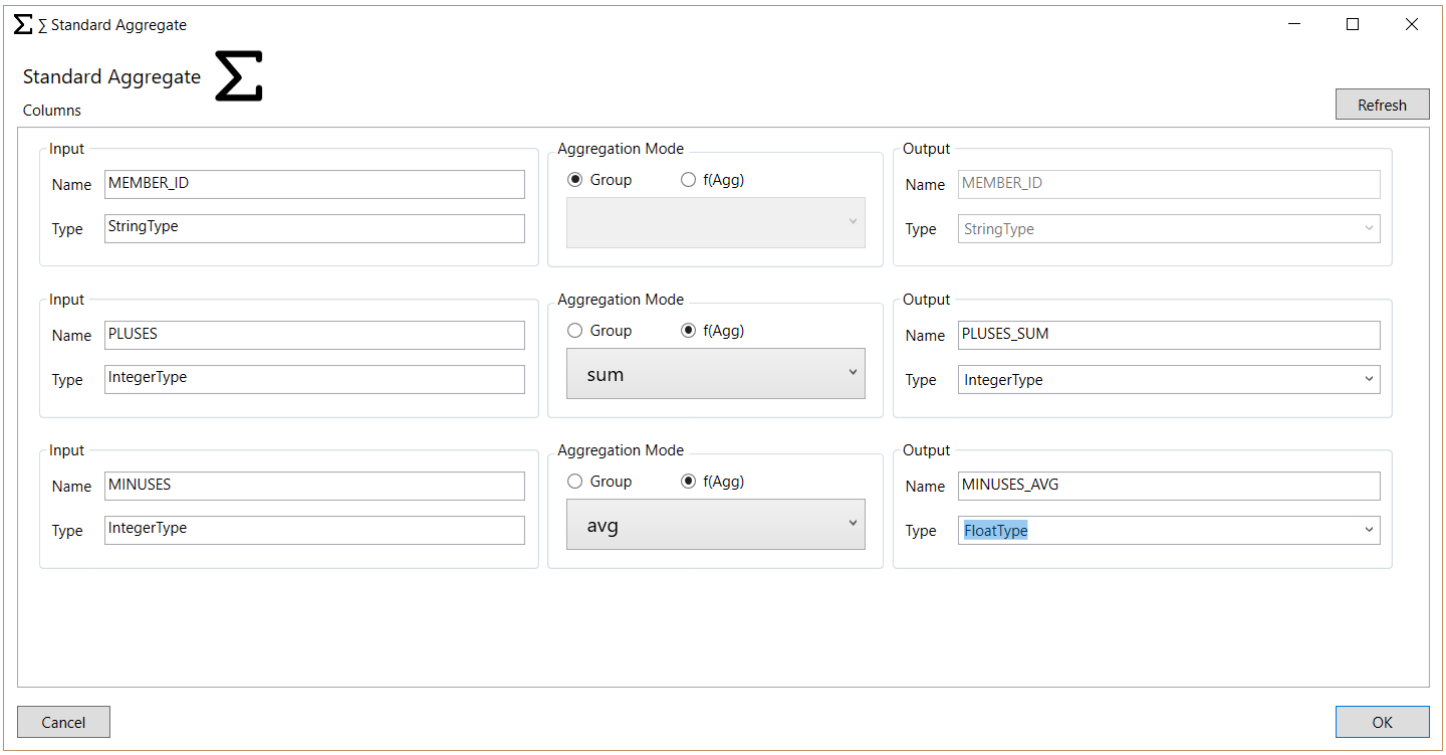
Example

Suppose the table MEMBER_SCORES contains the following data:

MEMBER_ID	PLUSES	MINUSES
1002	4	5
1002	5	6
1002	6	8
1003	6	6
1003	2	4
1003	4	7
1003	1	6
1006	4	3
1007	8	2

We will create a job to aggregate by MEMBER_ID, summing all of the PLUSES with the same MEMBER_ID and averaging all of the MINUSES with the same MEMBER_ID. The job looks like this:

The properties window for the Aggregate step will be as follows:



We choose Group for the column(s) (in this case, MEMBER_ID) that we are aggregating on, and f(Agg) for the columns that we are aggregating (in this case, summing on PLUSES and averaging on MINUSES). The interface suggests output column names and types for the aggregation columns, which you can edit.

In this case, the output will be as follows:

MEMBER_ID	PLUSES_SUM	MINUSES_AVG
1006	4	3.0
1007	8	2.0
1003	13	5.75
1002	15	6.3333335

Join

Introduction

The Join step performs SQL-type joins on datasets.

Example,

Suppose we have two tables, MEMBER_DATA and MEMBER_SCORES, containing the data

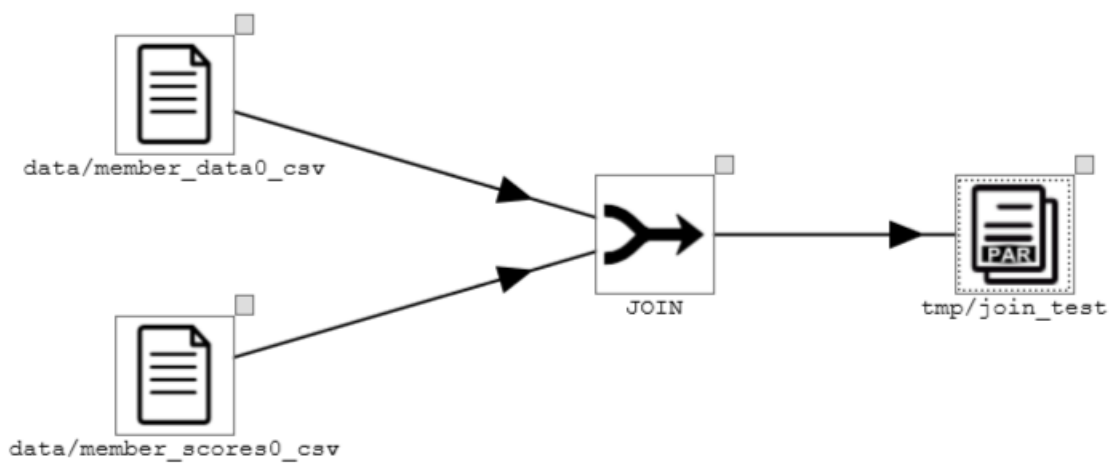
MEMBER_ID	MEMBER_NAME
1001	aaaa
1002	bbbb
1003	cccc
1004	dddd
1005	eeee
1006	ffff

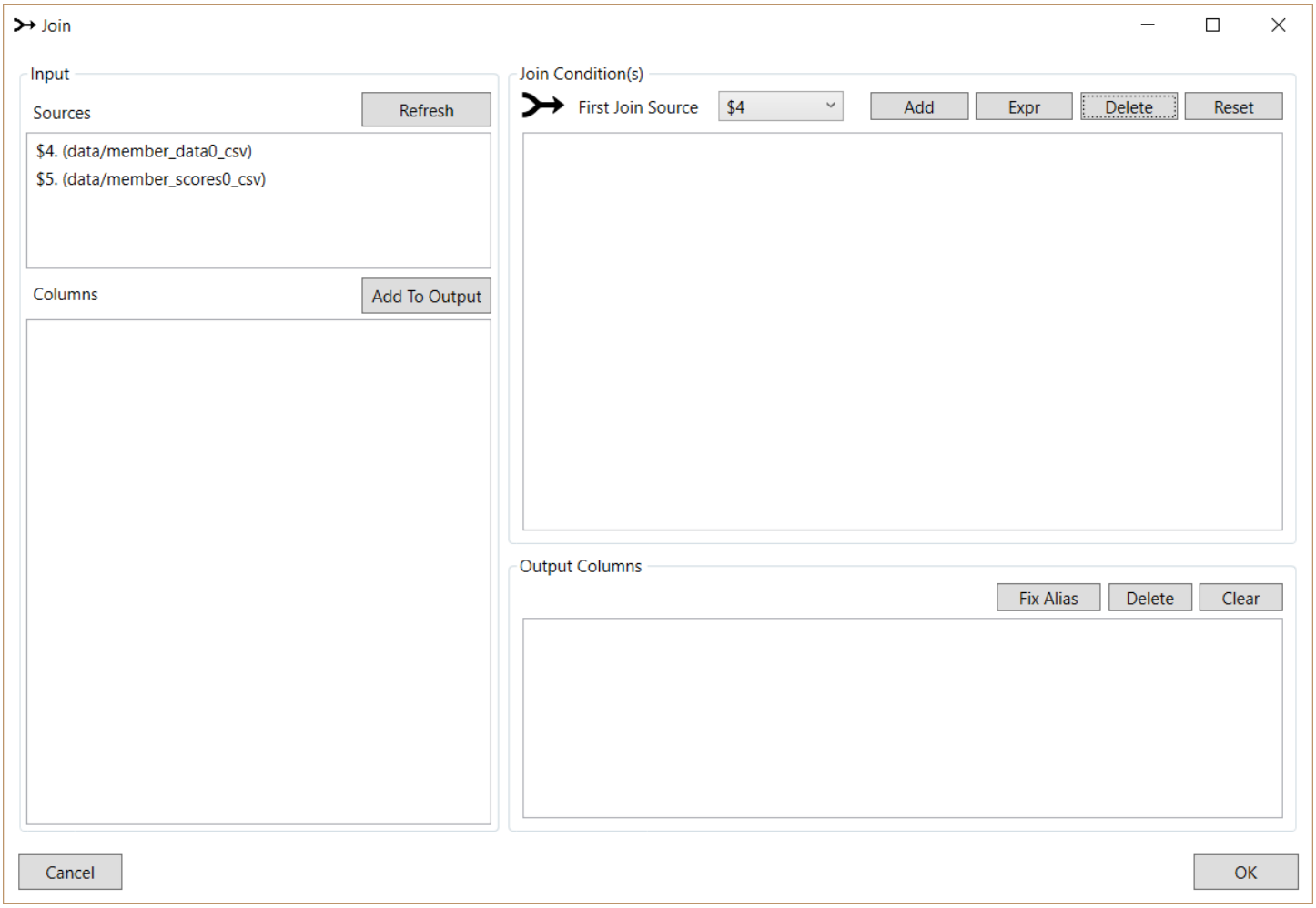
and

MEMBER_ID	PLUSES	MINUSES
1002	4	5
1003	6	6
1006	4	3
1007	8	2

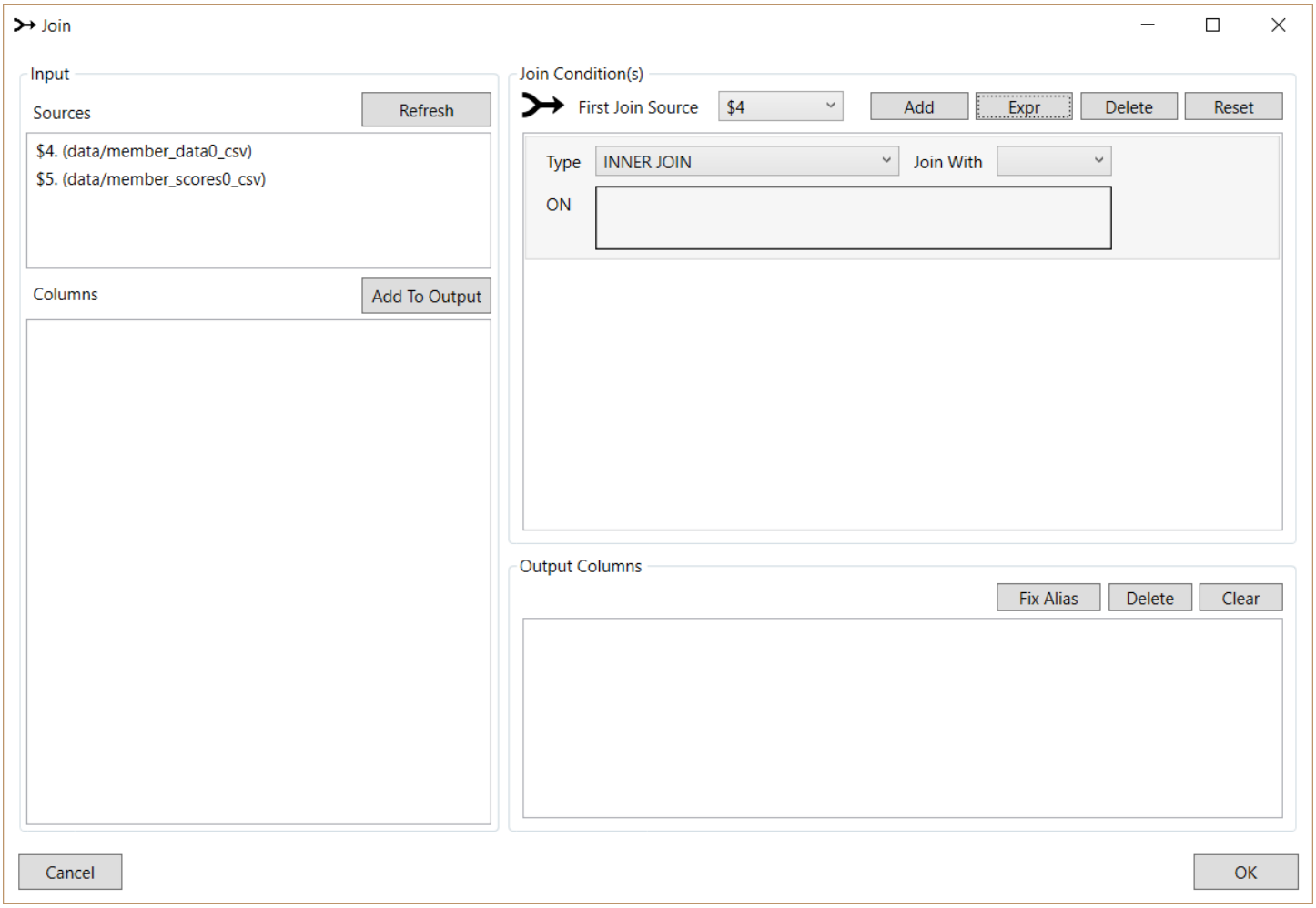
respectively.

We'll write a job to use the Join step to do an inner join on these two tables.



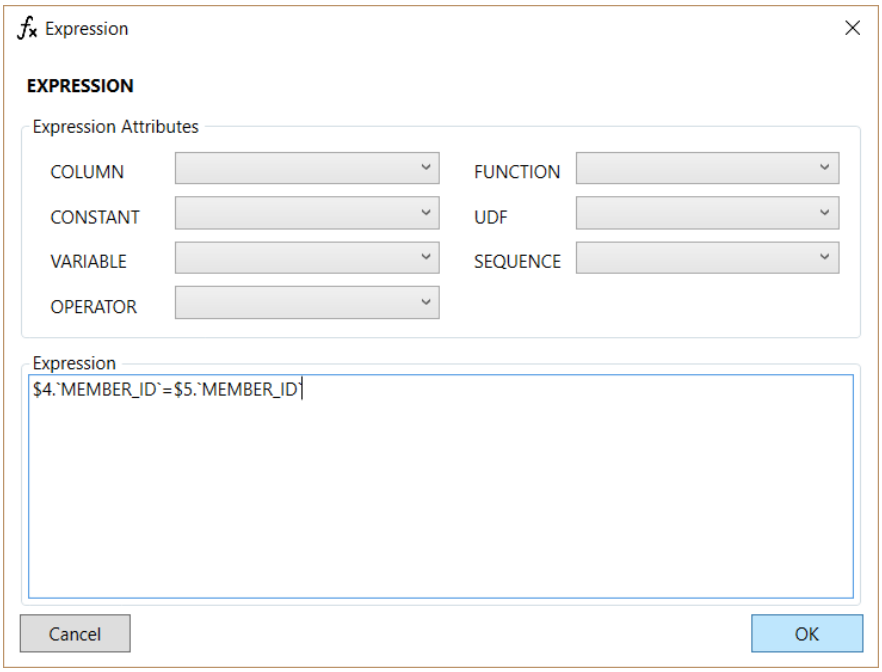


First, assign the role of First Join Source (or Left join source) to MEMBER_DATA, by choosing its number (in this case, $4) from the drop-down list. Then Click on [Add] to add a join expression:



Choose the join type from the Type drop-down list. In this case, we'll keep the default choice, INNER JOIN. Choose the number for MEMBER_SCORES in the Join With drop-down list. (This seems superfluous in this case, since there is only one other table, but Join can take more than two inputs, so in general, there may be more than one choice.). Now click on the Expr button.

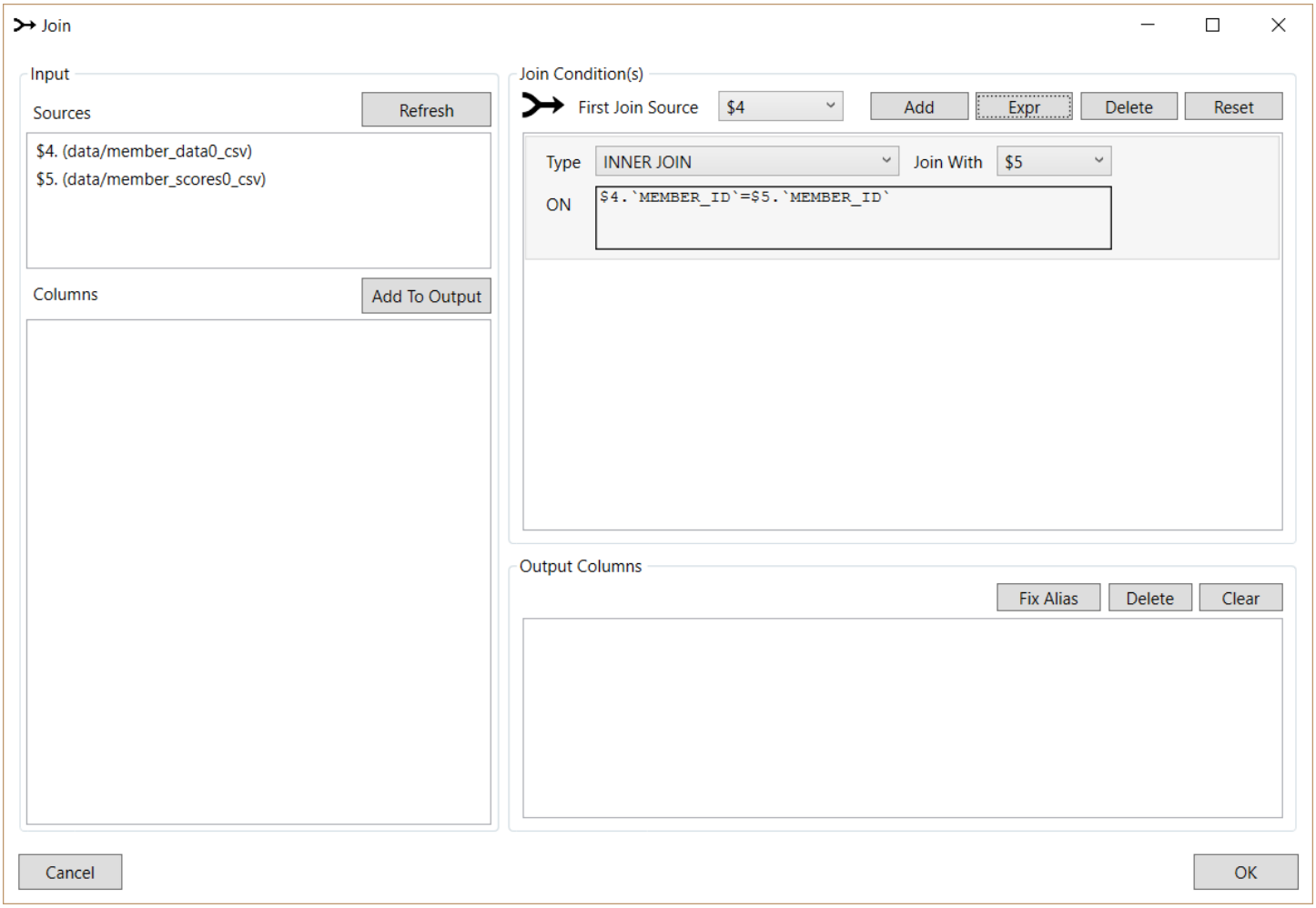
This will bring up an expression editor:



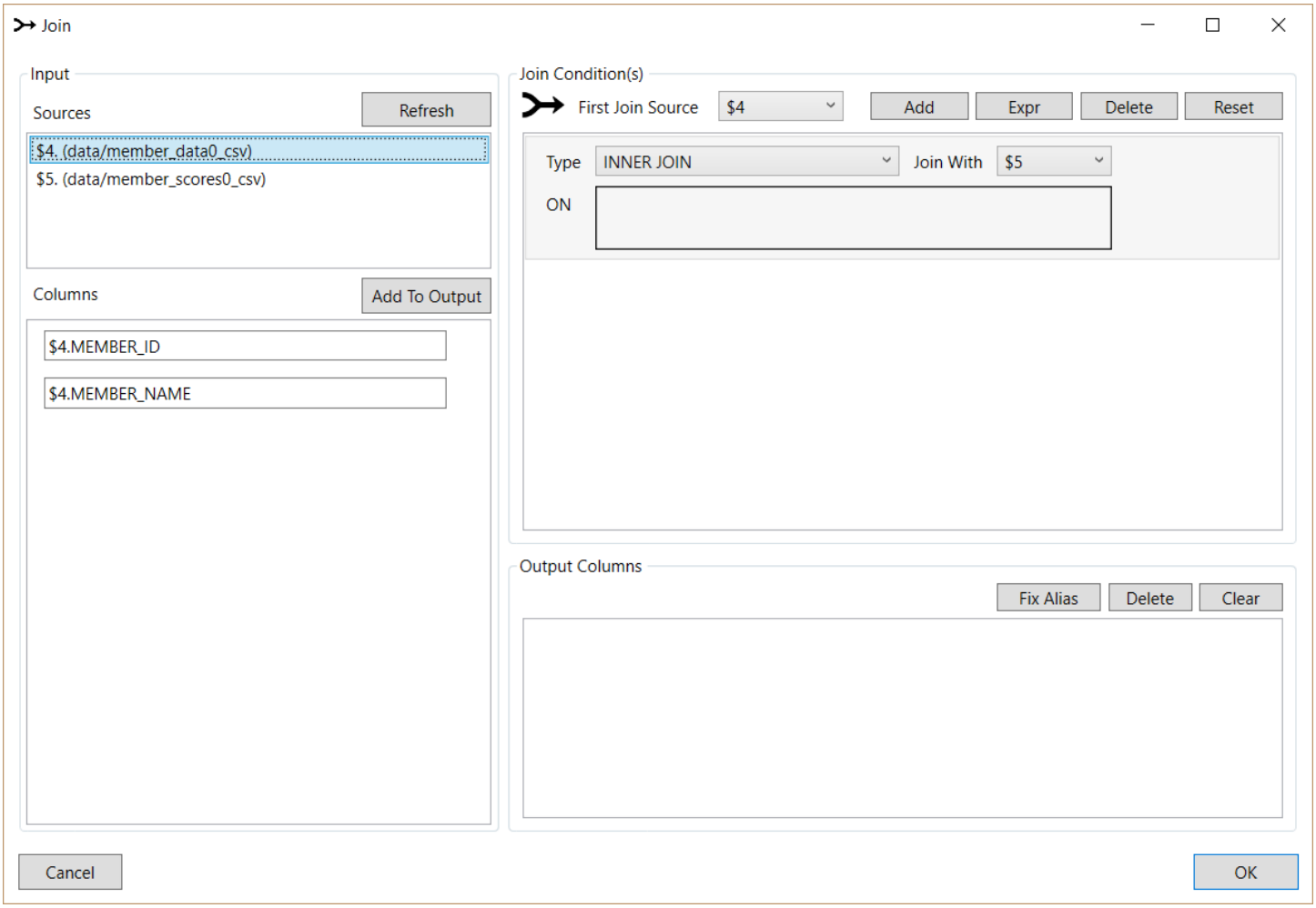
Enter an appropriate join expression in the editor – in our case, simply $4.`MEMBER_ID`=$5.`MEMBER_ID`. (Remember that column names must be surrounded by backquotes (`), as shown in the example. The COLUMN dropdown list will automatically supply them, but if you type the column names in yourself, the backquotes are your responsibility!)

Click [OK] to return to the properties window.

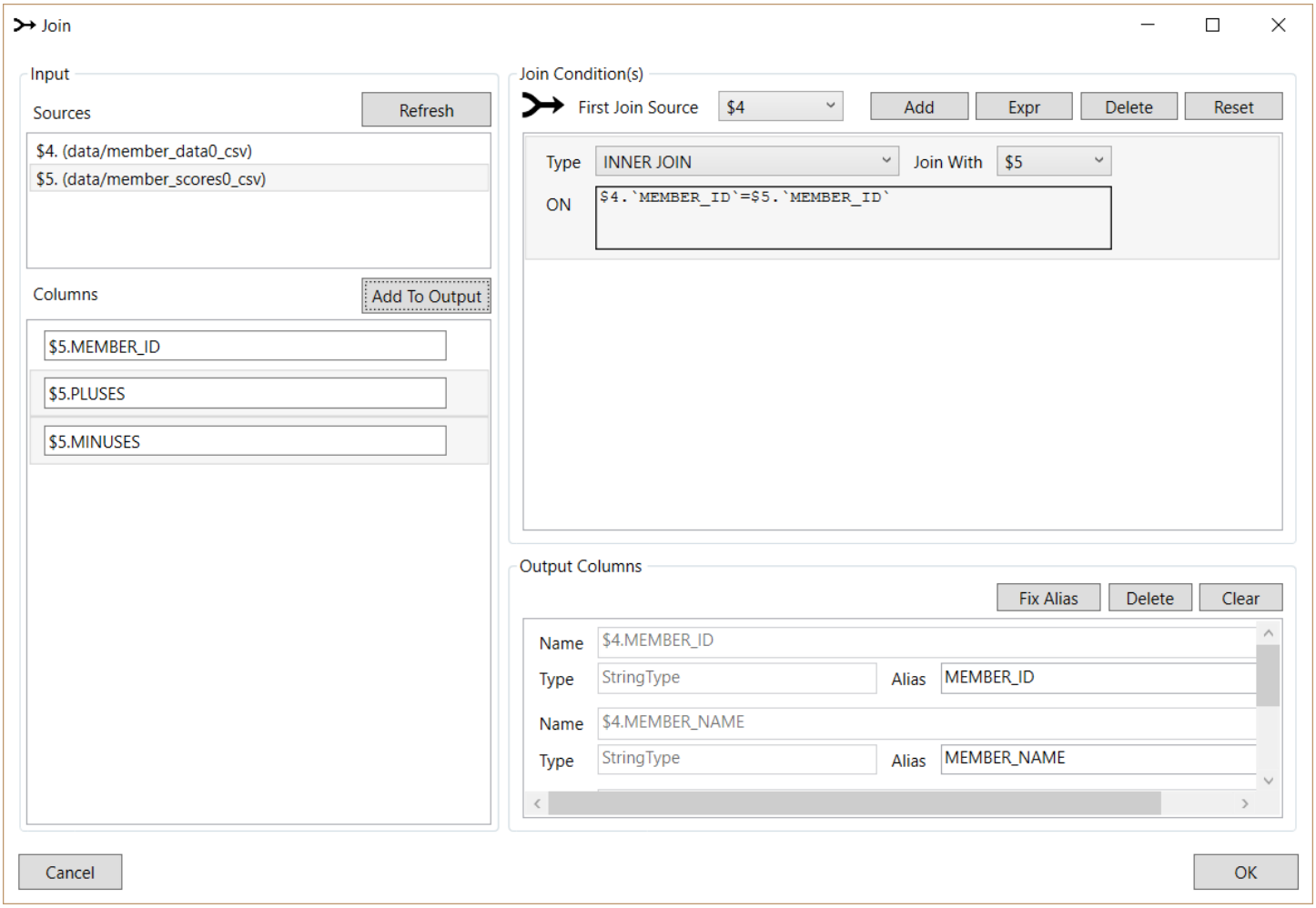
Now we must decide what columns get mapped from the input to the output.



The input tables are listed in the upper left-hand corner, under Sources. Clicking on each source will cause that source's columns to appear in the Columns section:



Then select the columns you wish to add to the output and Click the Add To Output button. You can select multiple columns at once by using the shift key. In our case, we'll add both MEMBER_ID and MEMBER_NAME from the MEMBER_NAMES dataset to the output. Then Click on the MEMBER_SCORES source and add PLUSES and MINUSES to the output.



Click [OK] to exit the properties window.

You will still need to complete the job by mapping the join output to an output stage and setting up the output stage by giving it the name of an output file and setting its other properties.

After running the job with the following input:

MEMBER_ID	MEMBER_NAME
1001	aaaa
1002	bbbb
1003	cccc
1004	dddd
1005	eeee
1006	ffff

and

MEMBER_ID	PLUSES	MINUSES
1002	4	5
1003	6	6
1006	4	3
1007	8	2

the output will be

```
+---------+-----------+------+-------+
|MEMBER_ID|MEMBER_NAME|PLUSES|MINUSES|
+---------+-----------+------+-------+
|     1002|       bbbb|     4|      5|
|     1003|       cccc|     6|      6|
|     1006|       ffff|     4|      3|
+---------+-----------+------+-------+
```

Had we chosen Join type of LEFT OUTER JOIN, the output would be

```
+---------+-----------+------+-------+
|MEMBER_ID|MEMBER_NAME|PLUSES|MINUSES|
+---------+-----------+------+-------+
|     1001|       aaaa|  null|   null|
|     1002|       bbbb|     4|      5|
|     1003|       cccc|     6|      6|
|     1004|       dddd|  null|   null|
|     1005|       eeee|  null|   null|
|     1006|       ffff|     4|      3|
+---------+-----------+------+-------+
```

The output for FULL OUTER JOIN would be

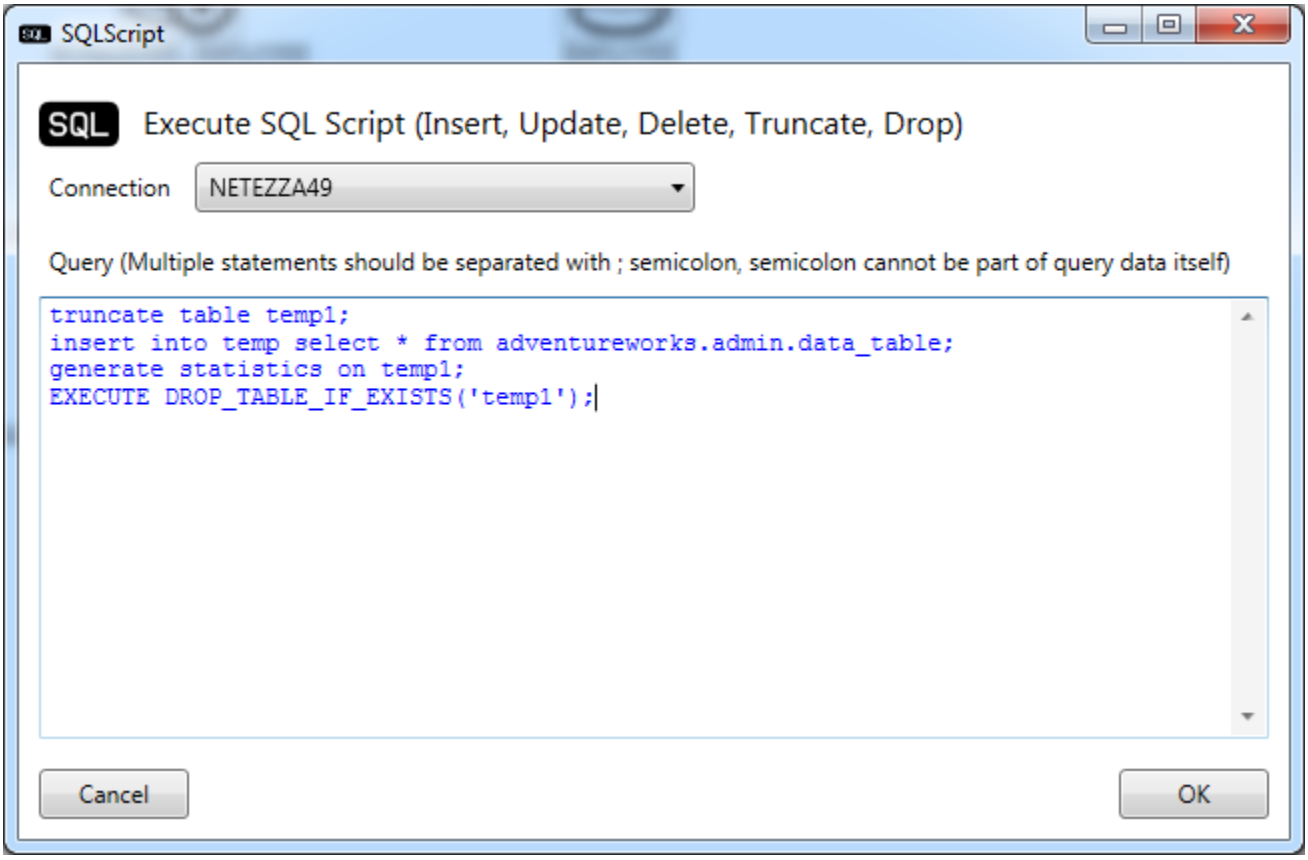
```
+---------+-----------+------+-------+
|MEMBER_ID|MEMBER_NAME|PLUSES|MINUSES|
+---------+-----------+------+-------+
|     1006|       ffff|     4|      3|
|     1003|       cccc|     6|      6|
|     1002|       bbbb|     4|      5|
|     1004|       dddd|  null|   null|
|     1005|       eeee|  null|   null|
|     1001|       aaaa|  null|   null|
|     null|       null|     8|      2|
+---------+-----------+------+-------+
```

SQL Script

Introduction

SQL Script step enables executing SQL queries. ***

Example,



SQL Statements

Multiple SQL statements are separated by semi-colon. Semi-colon cannot be part of query statement.
For example, using following statement will result in error during runtime.

```
insert into MY_DATA_TABLE select COLUMN1, COLUMN2||'added;' from
MY_SOURCE_DATA_TABLE
```

While above statement is still valid, runtime module splits entire content with semi-colon into multiple statements which leads to one or more invalid SQL.

Calling Stored Procedures

Executing procedures in SQLScript step is also possible which can be achieved by simply inserting call procedure statement. Stored procedure calls only work on your targeted platform connection.

Example,

```
EXECUTE DROP_TABLE_IF_EXISTS('temp1');
```

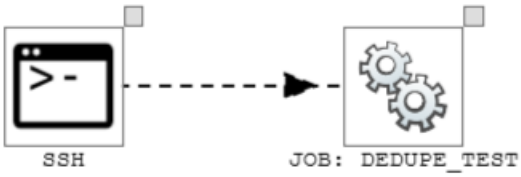
SSH

Introduction

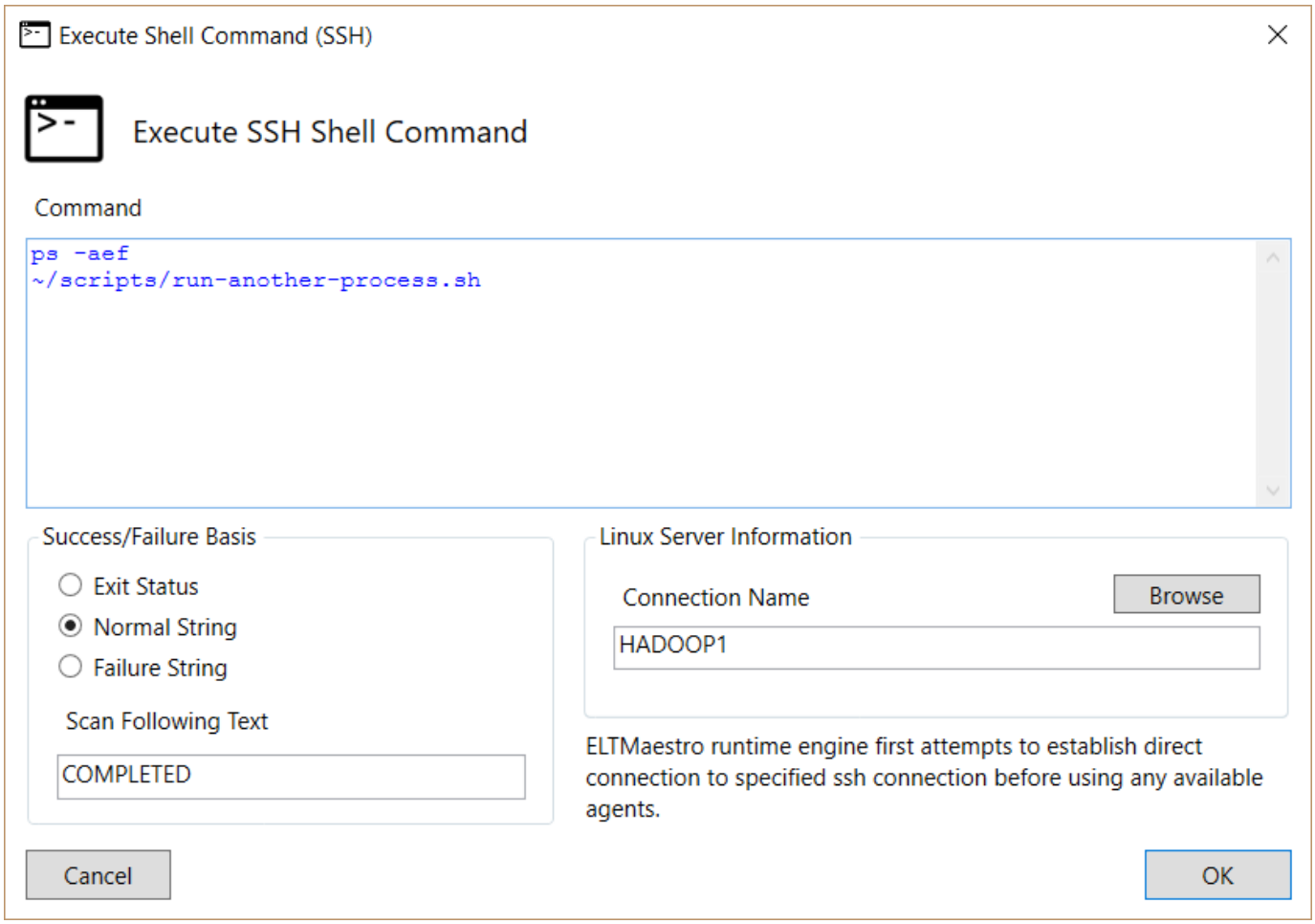
The SSH step runs a shell command on one of the Spark nodes. You can use the result of the shell command to control the execution of a subsequent step.

Example

We set up our job as follows:



In the properties of the SSH command, we enter the following:



where run-another-process.sh is a shell script that may or may not echo the string 'COMPLETED' to standard output. When the script indeed produces the string 'COMPLETED', the step is successful, which then causes the DEDUPE_TEST job to run; this action can be observed on the job canvas with Debug mode enabled:

.

If we alter the script so that it no longer echoes 'COMPLETED', the SSH step fails, and the DEDUPE_TEST job does not run.

<u>Success or Failure Options</u>

Exit Status

If exit status code of shell script equals zero then set this step status to SUCCESSFUL, otherwise FAIL.

Normal String

If output contains mentioned string to be scanned set step status to SUCCESSFUL, otherwise FAIL.
Exit status is ignored if this option is checked.

Failure String

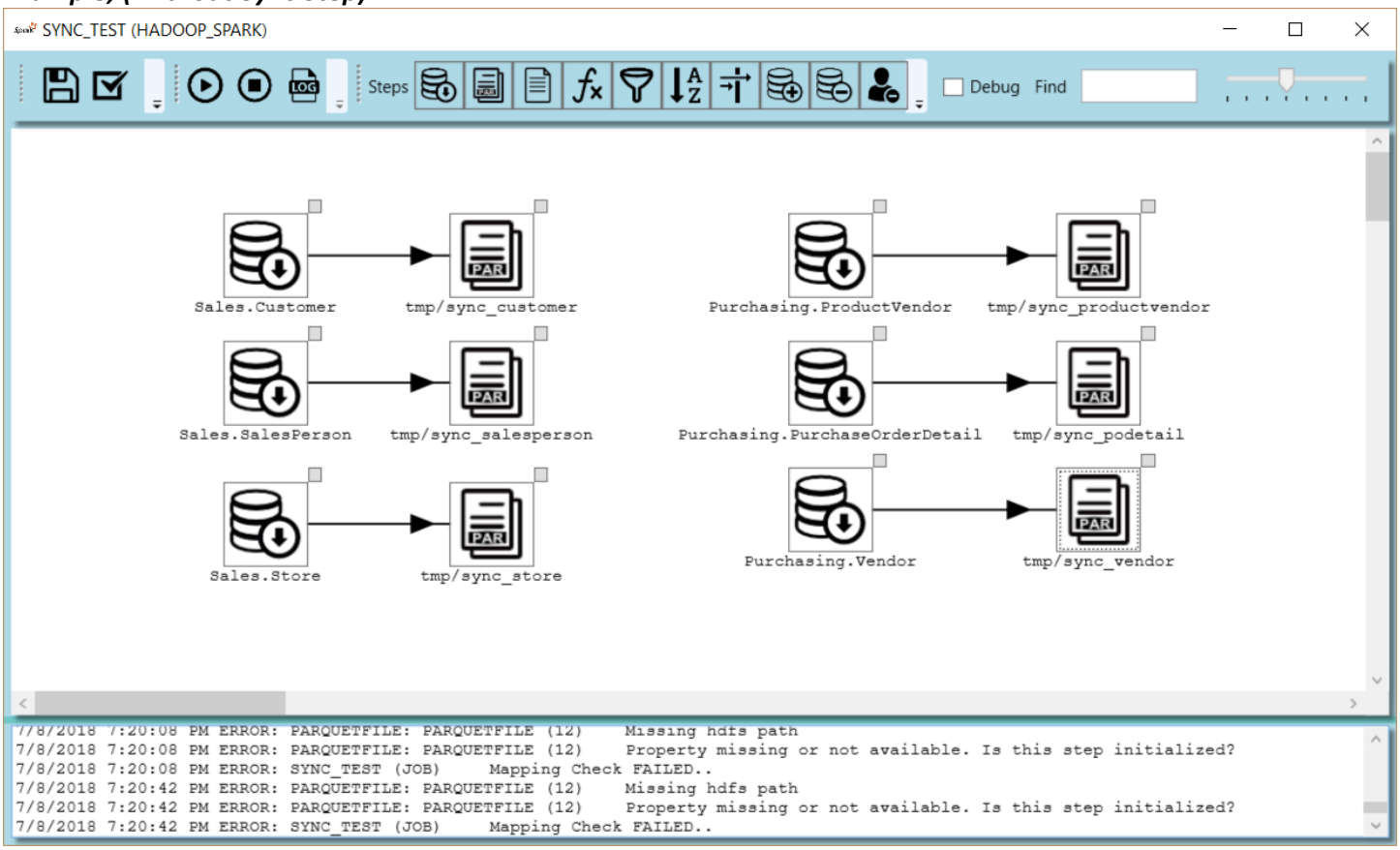
If output of script contains mentioned string to be scanned set step status to FAIL, otherwise SUCCESSFUL.
Exit status is ignored if this option is checked.

Sync

Introduction

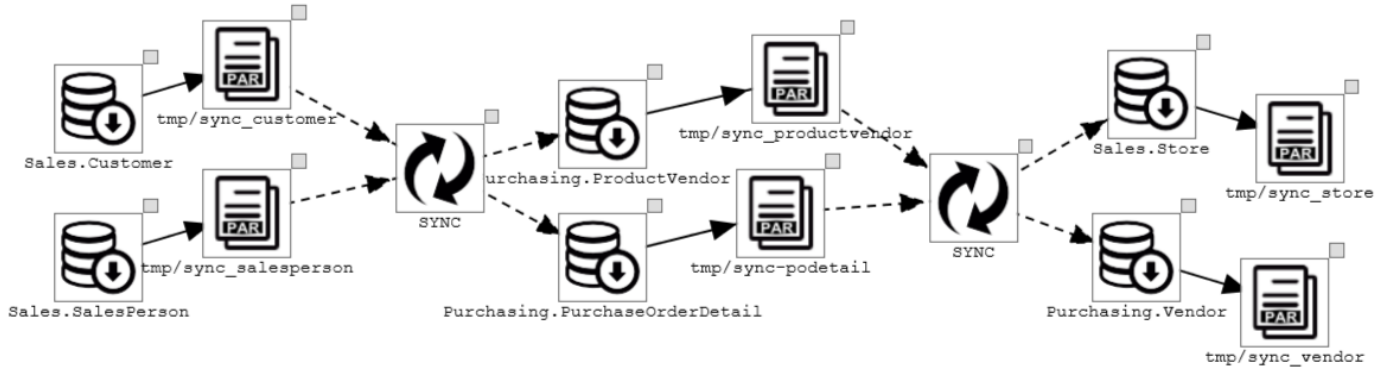
Sync is a dummy step which always returns successful state if executed. The purpose of this step is to control ELT parallelism for certain MPP systems that contain high throughput and low concurrency. Sync step does not have any configuration property.

Example, (Without Sync Step)



Above workflow executes 6 workflow paths in parallel, which means the number of connections on source and target amounts to 12 (6 on source, 6 on target). This number can easily go very high when adding more tables. To implement phasing mechanism which is to load few, wait, load more, repeat kind of operation Sync step should be considered when designing parallel processes.

Example, (With Sync Step)



Above workflow executes in following order:

1. Load two tables in parallel.
2. Sync waits for completion.
3. Load two more in parallel.
4. Sync waits for completion
5. Load two more in parallel.

Switch

Introduction

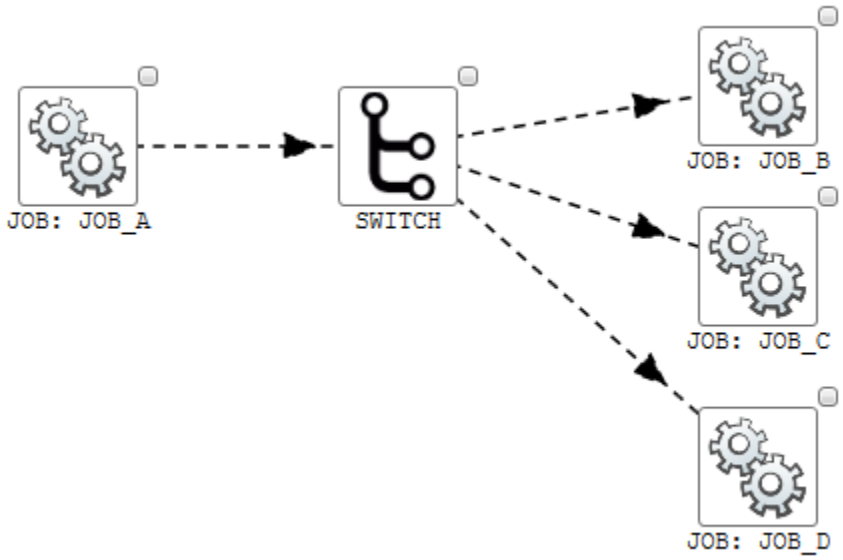
Branch success/failure paths based on input step state.

Example,

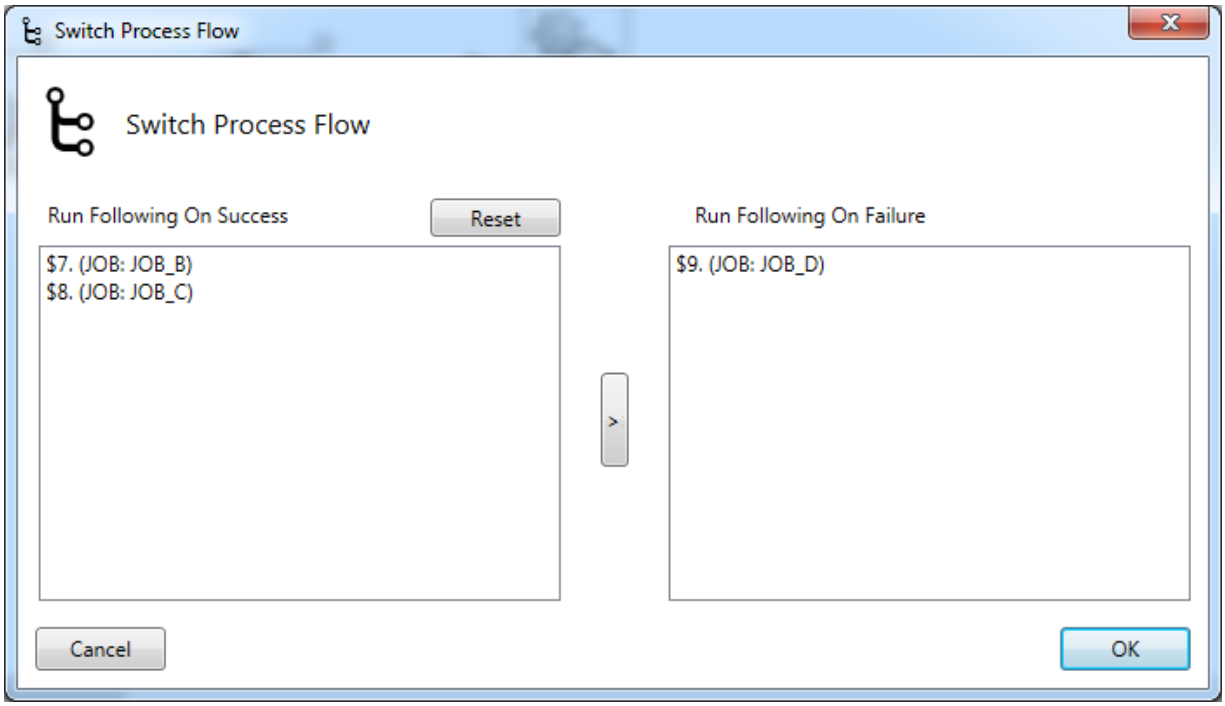
Sample dependency logic

1. Run JOB_A
2. If JOB_A succeeds Run JOB_B and JOB_C (Skip Failure Path)
3. If JOB_A fails Run JOB_D (Skip Success Path)

Workflow should look something like following.



Switch step property example.

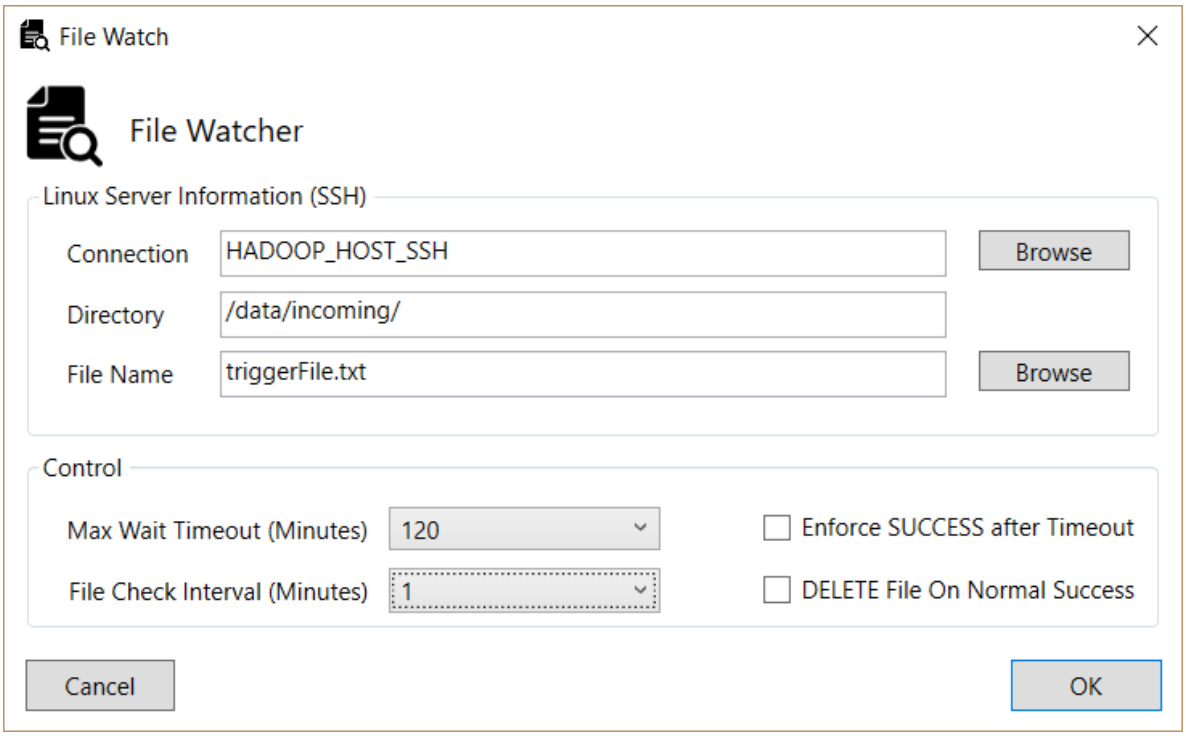


FileWatch

Introduction

FileWatch step waits until certain file on Unix/Linux system becomes available until timeout.

Example



Linux Server Information

Linux server ssh credentials. Directory to be scanned and filename to watch for.

Max Wait Timeout

Wait for certain minutes until timeout has occurred.

File Check Interval

Interval to Check availability of file.

Enforce SUCCESS after Timeout

If option is checked step status will not fail after timeout. Leaving unchecked sets step status to Failed if file is not found and timeout has occurred.

Delete File On Normal Success

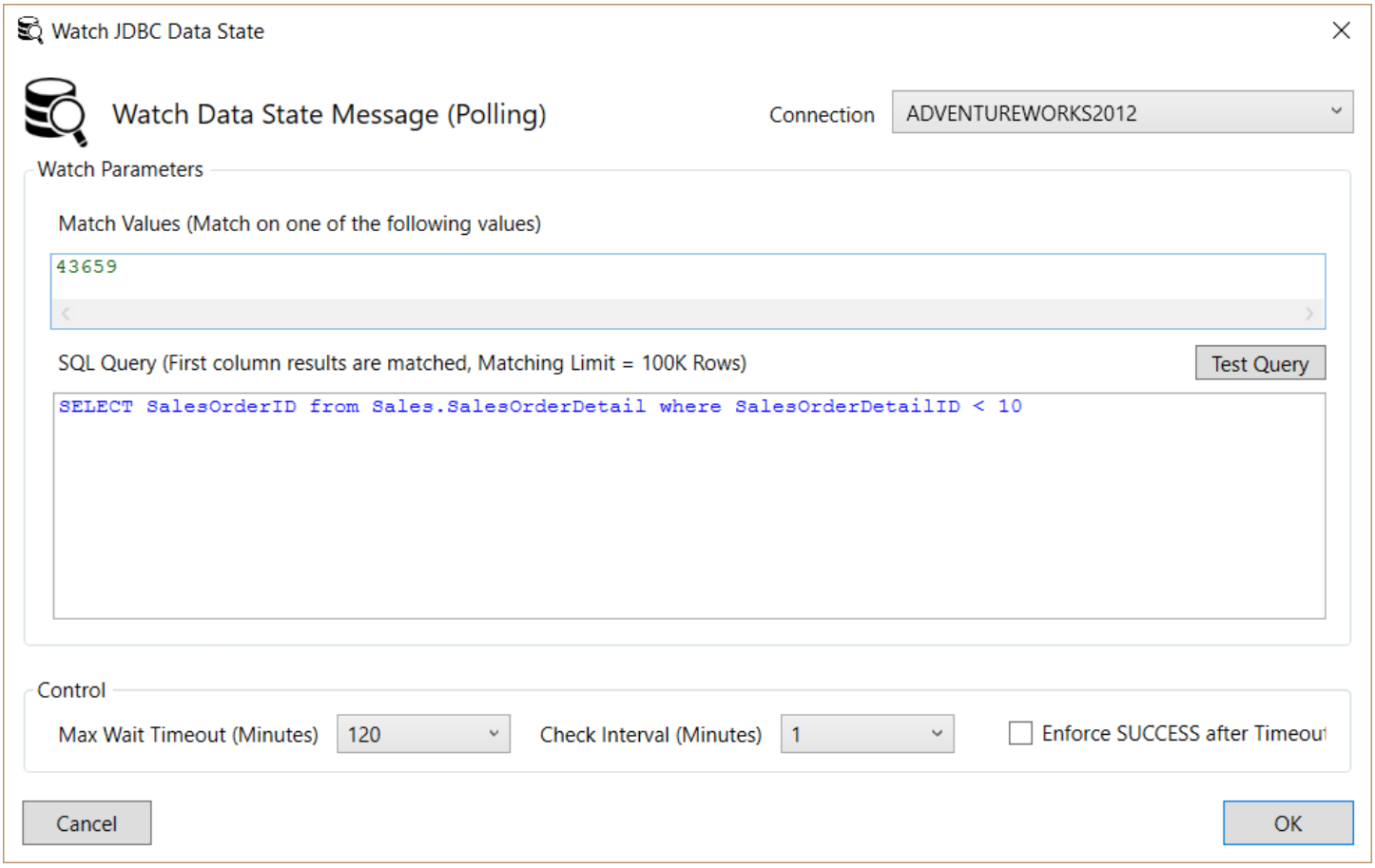
If checked, file gets deleted after setting status to success.

JDBCWatch

Introduction

JDBCWatch step waits until query returns certain value on a specified connection.

Example,



Match Values

Output to be matched against. True if one of the values match.

SQL Query

SQL query to collect output. State is success if any result tuple matches any specified match value (First row-column value).

Max Wait Timeout

Wait for certain minutes until timeout has occurred.

Check Interval

Interval to repeat query.

Enforce SUCCESS after Timeout

If option is checked, step status will not fail after timeout. Leaving unchecked sets step status to failed if match is not found.

Set Variable

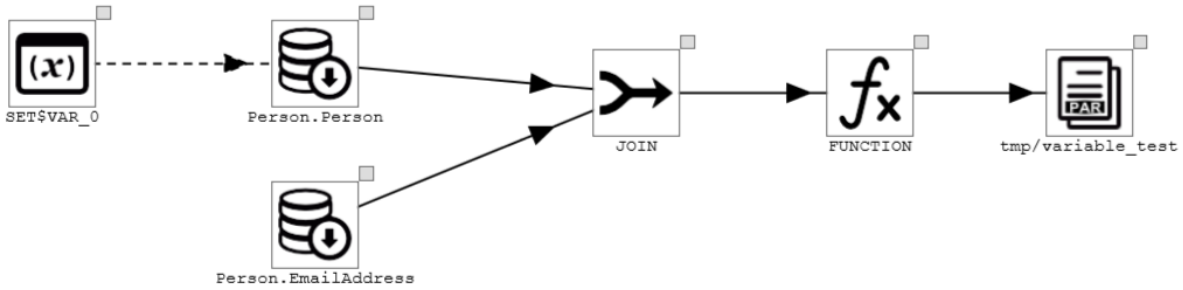
Introduction

The Set Variable step adds the ability to communicate between a Unix shell or an SQL database and an ELTMaestro program.

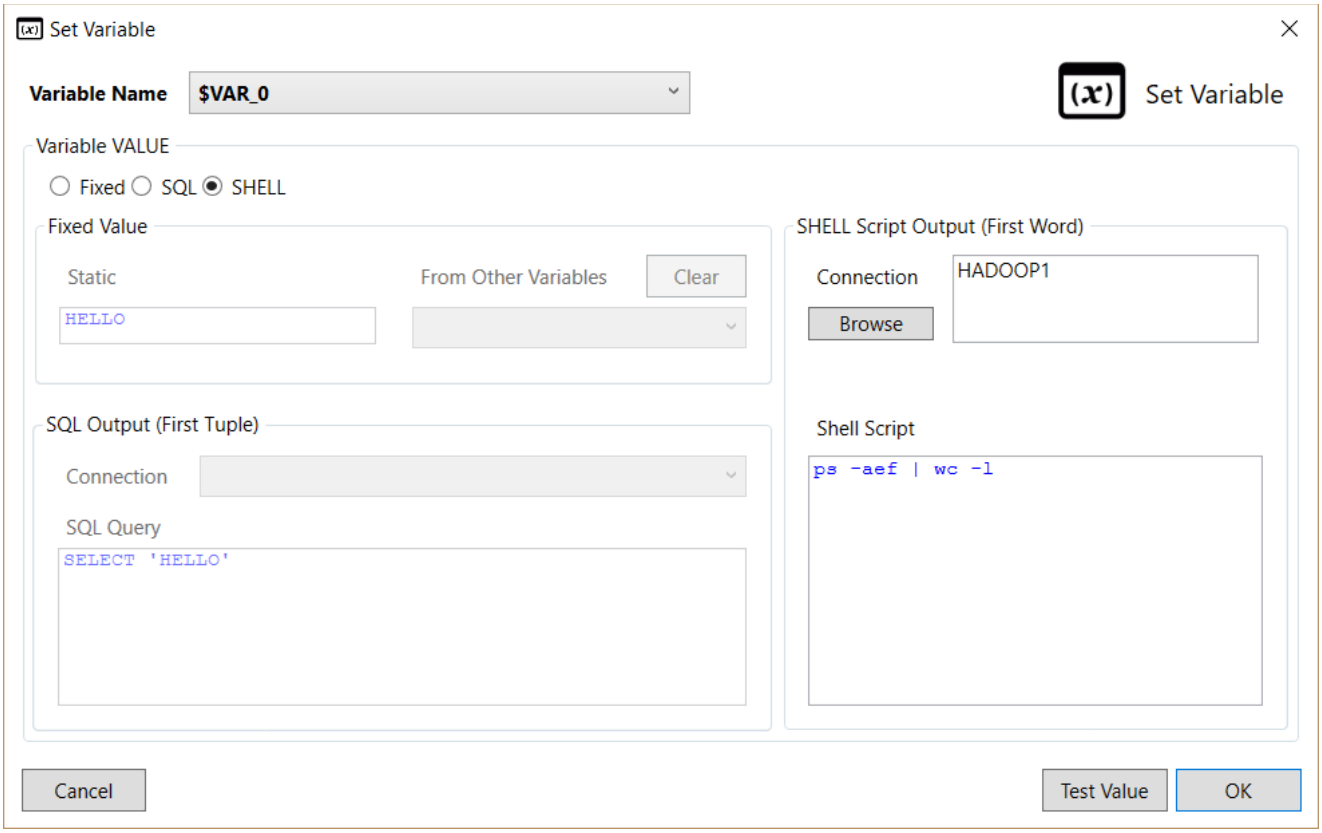
Each job is associated with a number of user variables, named $VAR_0, $VAR_1, $VAR_2, … (in addition to a number of system variables such as $JOB_NAME, $JOB_ID, etc.). The user variables are set in the Set Variable stage and accessed by various other stages. Variables facilitate communication between different parts of ELTMaestro programs. Because the Set Variable stage allows variables to have their values set by interaction with Unix shells and databases with which ELTMaestro can establish a connection, the Set Variable stage also facilitates communication between ELTMaestro processing Unix shells or databases of interest.

Example,

In this example, we use the Set Variable step to track the number of processes on one of the machines in the Spark cluster while the job is running and write that number to a column in an output table. Here we use a simple Join job, similar to the one used to demonstrate the Join command above. Next we connect the Set Variable step to one of the OnStage input steps at the beginning of the program, as shown below:

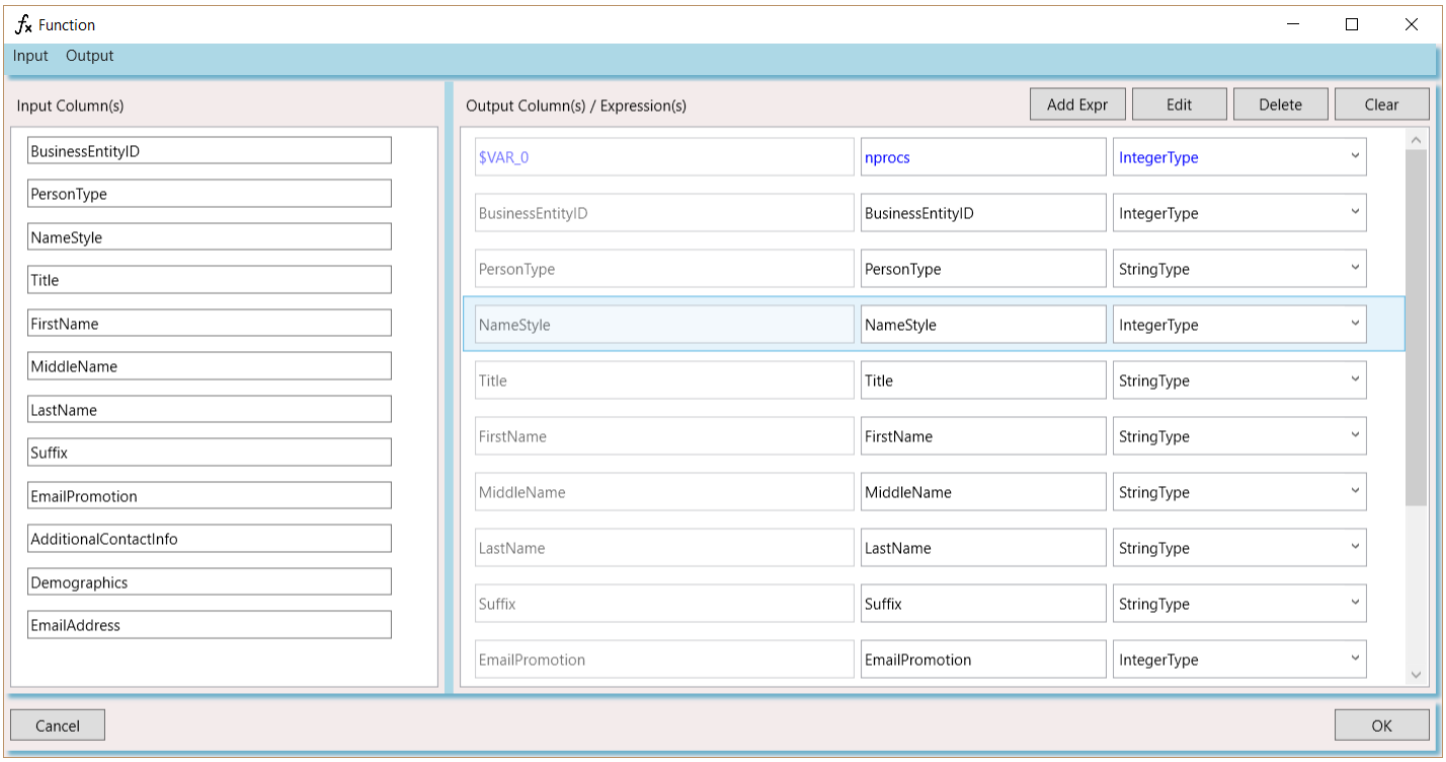


Now open the Set Variable step:



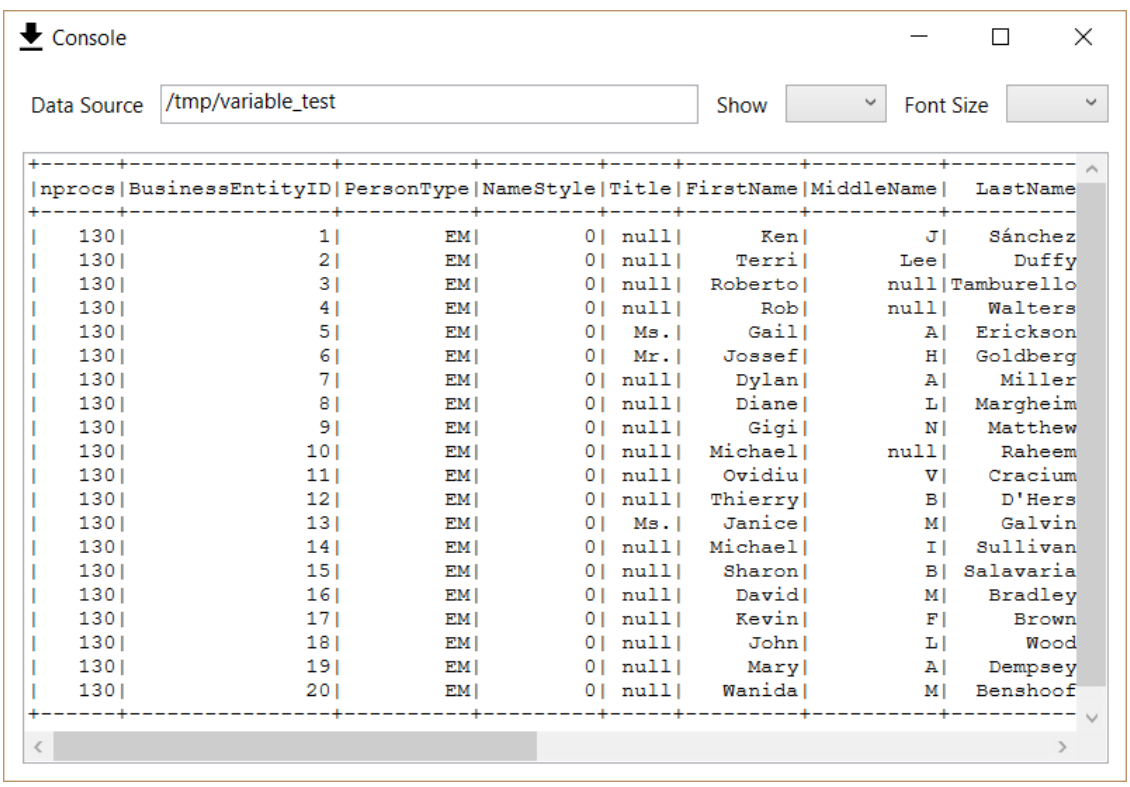
In the Variable Name dropdown list, we have selected $VAR_0 – in other words, $VAR_0 is the particular variable we are setting.
From among the Fixed, SQL, and SHELL radio buttons, we have selected SHELL, indicating that the value that winds up in $VAR_0 will come from a shell script.
HADOOP1 happens to be the name of the Unix machine we connect to. (That would likely be different in your case.)
The shell script contains a short program to count the lines produced by the command **ps -aef**.

Note that we have also introduced a Function step between the Join step and the output file. The Function step allows us to retrieve the value of the variable and add it to the dataflow. Let's take a look inside the Function step:



Here we see that we have added a new output column, named **nprocs**, of type integer, and set it to $VAR_0.

After running the program the results appear as follows:



nprocs	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName
130	1	EM	0	null	Ken	J	Sánchez
130	2	EM	0	null	Terri	Lee	Duffy
130	3	EM	0	null	Roberto	null	Tamburello
130	4	EM	0	null	Rob	null	Walters
130	5	EM	0	Ms.	Gail	A	Erickson
130	6	EM	0	Mr.	Jossef	H	Goldberg
130	7	EM	0	null	Dylan	A	Miller
130	8	EM	0	null	Diane	L	Margheim
130	9	EM	0	null	Gigi	N	Matthew
130	10	EM	0	null	Michael	null	Raheem
130	11	EM	0	null	Ovidiu	V	Cracium
130	12	EM	0	null	Thierry	B	D'Hers
130	13	EM	0	Ms.	Janice	M	Galvin
130	14	EM	0	null	Michael	I	Sullivan
130	15	EM	0	null	Sharon	B	Salavaria
130	16	EM	0	null	David	M	Bradley
130	17	EM	0	null	Kevin	F	Brown
130	18	EM	0	null	John	L	Wood
130	19	EM	0	null	Mary	A	Dempsey
130	20	EM	0	null	Wanida	M	Benshoof

That is, there were evidently 130 processes running on HADOOP1 as the Join job shown above ran.

Variable Name

Name of variable defined while creating job.

Variable Values

Fixed :	Can be static value or copy from another variable.
SQL Output:	Output of SQL query on specified connection is used to load variable value. First tuple (first row-column) is used from query result.
Shell Script:	Output of shell script is used to load variable value. First word displayed on standard output is selected.

Note

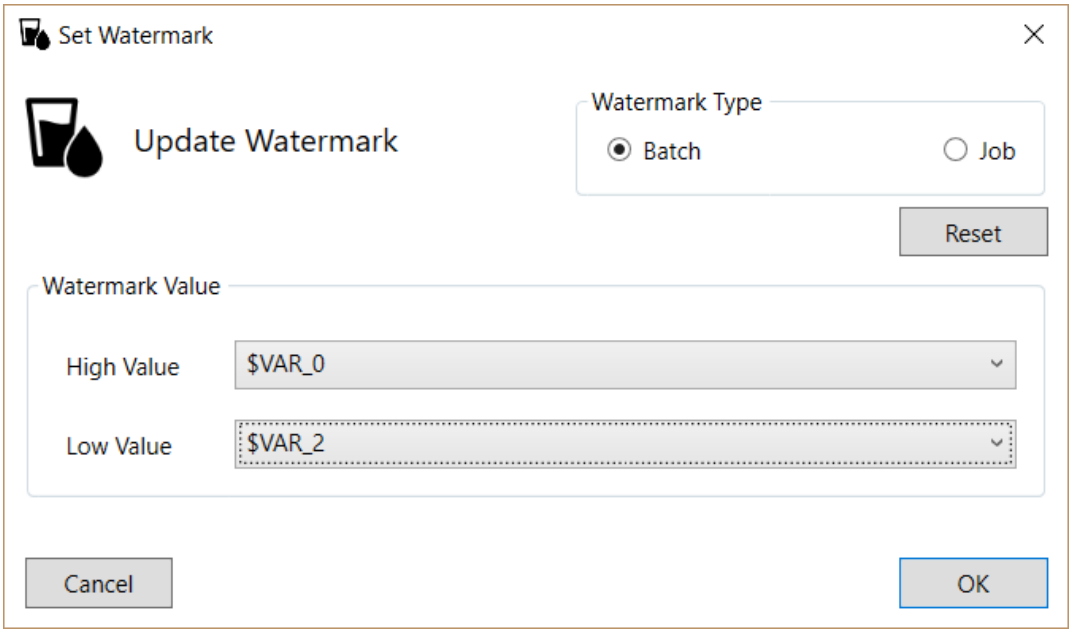
All variable values are re-evaluated during runtime.

WaterMark

Introduction

Sets workflow JOB or root job (batch) watermark from another variable. Watermark values can only be set by copying from pre-initialized variables (current job variables).

Example,



Watermark Types

Batch:	Root job watermark.
Job:	Current job watermark.

Watermark Values

High Value:	High watermark value for checked watermark type.
Low Value:	Low watermark value for checked watermark type.

Note

When a workflow runs following watermark values are initialized automatically by the engine. The watermark values are captured from last successful run state.
$BATCH_LOW_WATERMARK_VALUE
$BATCH_HIGH_WATERMARK_VALUE
$JOB_HIGH_WATERMARK_VALUE
$JOB_LOW_WATERMARK_VALUE

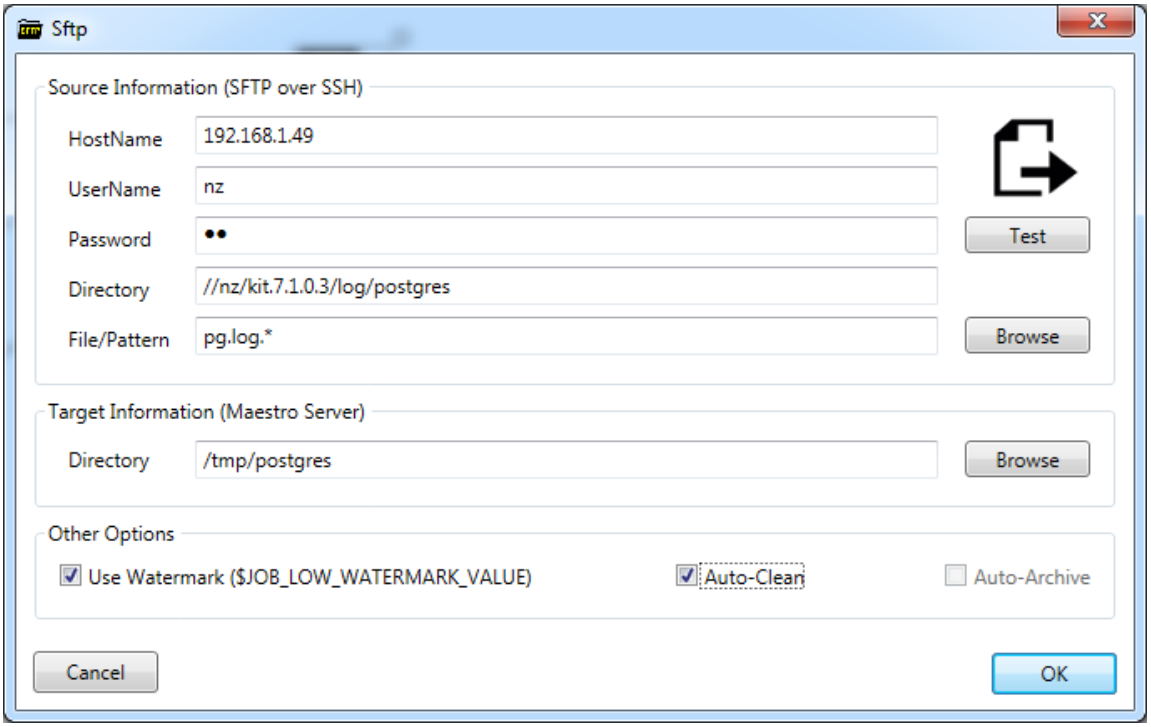
Browse to **Variables and Watermark** section for more information.

SFTP

Introduction ***

SFTP step enables downloading files from remote UNIX/Linux servers. ELTMaestro connects using SSH protocol to retrieve files using secure channel. SFTP step can also utilize watermarks to enable downloading changed files.

Example,



Source Information

SSH Login:	SSH credentials for SFTP server.
Directory:	Source Directory
File/Pattern:	File Name or Pattern. POSIX expression is used to evaluate file names.

Target Information

Directory :	Directory path on ELTMaestro server.

Use Watermark Option

If this option is checked ensure that current workflow does not set Job Low Watermark Value. Watermark option utilizes job low watermark value to obtain only the changed files since last load. SFTP step automatically keeps track of latest file modified timestamp based on source server time zone and updates job low watermark value automatically.

Auto Clean Option

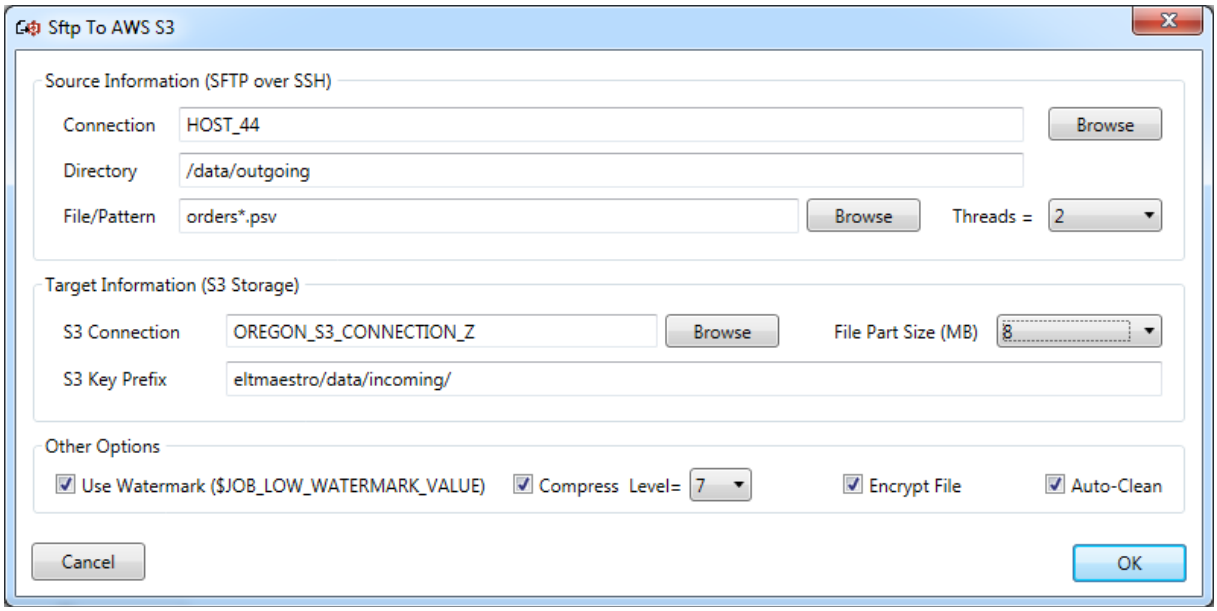
If this option is checked upon completion of root job (BATCH) the downloaded files are automatically deleted. Auto-Clean option is useful specially when freeing up disk resources on ELTMaestro server after loading them into database tables.

SFTP2S3

Introduction ***

SFTP2S3 step enables downloading files from remote UNIX/Linux servers into one of the active agent systems and then uploading those files to S3 bucket. ELTMaestro connects using SSH protocol to retrieve files using secure channel. SFTP step can also utilize watermarks to enable downloading changed files. File(s) can be encrypted and/or compressed before uploading into S3 bucket. Client side AES256 bit encryption is default encryption algorithm.

Example,



Parameters

Property	Type	Info
Connection	Text	SSH Connection Name
Directory	Text	Source Directory
File / Pattern	Text	Filename or POSIX Pattern
Threads	Selection	Number of threads for parallel uploads. (Applies to pattern matched files)
File Part Size (MB)	Selection	S3 upload file partition size for larger files.
S3 Connection	Text	S3 Connection Name
S3 Key Prefix	Text	Key Prefix to append
Use Watermark	Check Box	Optional: Uses low watermark variable.
Compress	Selection	Optional: Uses bzip2 compression, 9 is highest compression level
Encrypt File	Check Box	Encrypts file before uploading using specified key configured on S3 Connection. Uses AES256 bit client side symmetric key.
Auto-Clean	Check Box	Deletes uploaded files from S3 upon job completion.

Use Watermark Option
If this option is checked ensure that current workflow does not set Job Low Watermark Value. Watermark option utilizes job low watermark value to obtain only the changed files since last load. Step automatically keeps track of latest file modified timestamp based on source server time zone and updates job low watermark value automatically.

Auto Clean Option
If this option is checked upon completion of root job (BATCH) the downloaded files are automatically deleted. Auto-Clean option is useful specially when freeing up disk resources on ELTMaestro server after loading them into database tables.

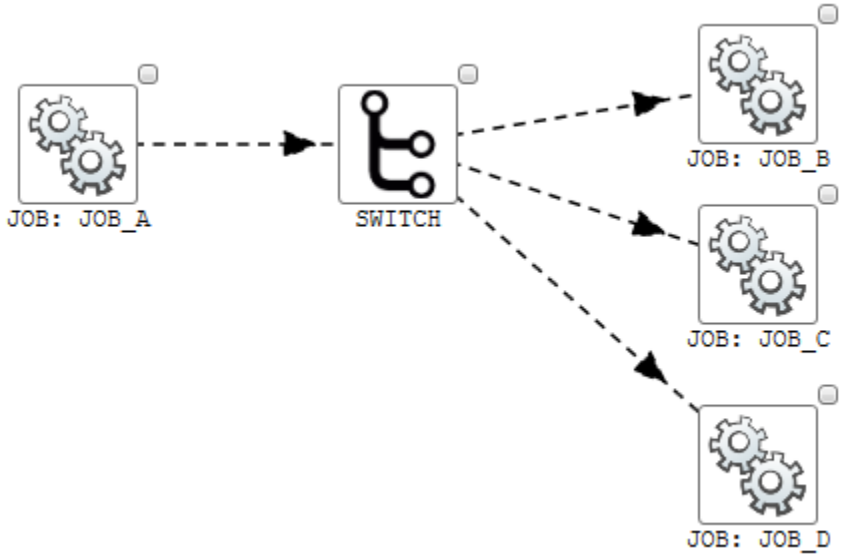
Note: Flat File Step Awareness
This is automatic. If this step is connected to flat file step, the uploaded file names are passed to target step for loading. Flat file step must select SFTP Step as source in order to function properly.

JobStep

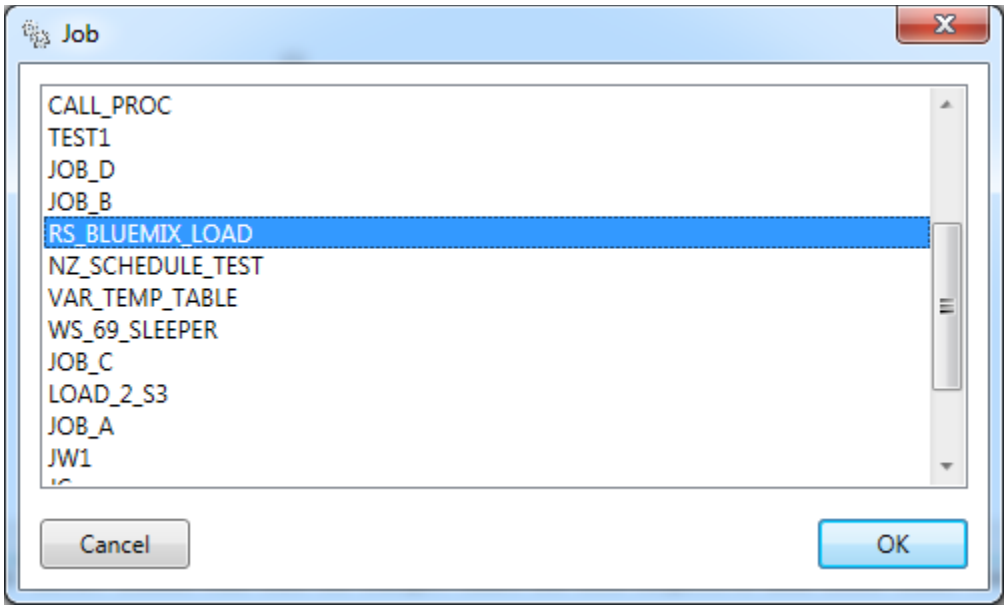
Introduction

Allows current workflow to execute deployed workflow. JobStep can be used with Switch to design success and recovery workflow path as well.

Example Workflow Implementation,



Example,



Job

Job Step executes selected deployed job.

Note

ELTMaestro engine can only execute one instance of a job in a workflow to avoid execution recursion.

Appendix A. ABC Database Tables

Table	Description
JOB	This table contains Information on data integration processes, their names and what they do. A job is synonymous with a process.
CONTROL_TEST	This table defines the balancing tests that need to be run periodically to check the correctness and quality of one or more data integration processes.
JOB_SOURCE_TARGET	This Table defines the data sources and data target tables and files used in a data integration job or process.
BATCH_CYCLE_TYPE	Contains information of whether the batch is real-time continuous, daily, weekly, monthly, etc.
SOURCE_TARGET_TYPE	Provides a classification of sources and targets such as STAGING, FACT, DIMENSION, OLTP System, EW, etc.
CONTROL_TEST_HIERARCHY	This table is used to define a hierarchy of tests that eventually roll up to subject areas and ultimately to the enterprise warehouse as a whole. The hierarchal structure is directly reflected in the organization of the "stop light" report shown in figure 2.16 which is presented to the larger user community upon login to the BI reporting subsystem.
CONTROL_TEST_POINT	This table associates balancing/data quality tests, to the source tables used to define the expected values and the target tables that are the sources of the actual value measurements. Generally, all CONTROL TESTS, except those with a CONTROL_TEST_TYPE of 'P' or 'S' or 'M' will have at least two test points.
CONTROL_TEST_TYPE	Defines the Type of Balancing/Data Quality test or measurement performed. See Appendix D for a description of valid test types.
BATCH_CYCLE	This table holds the definitions of groups of data integration and data quality/balancing processes that are run together as a batch at some predefined interval such as continuously, Friday of each week, the last day of each month, etc. Note the ABC database does not run any batches; it only records metadata about the batches, the jobs within a batch and what happened when they ran.
BATCH_CYCLE_JOB	This table groups data integration and data quality/balancing tests together into a batch that are run together in a predefined order.
HOST	This table holds definitions of the computing hosts (systems) that contain the files or databases used as data sources and targets. This data is informational only

	and used as a quick reference when troubleshooting or tracking down data integration or data quality issues.
DATA_SOURCE_TARGET	Contains an entry for each data object, its classification and what host it is found on.

The following Tables are populated during a batch cycle run. Records will be inserted during the initiation of the data integration process and updated during course of process execution.

Table	Description
BATCH_CYCLE_RUN	This table uniquely identifies and records metadata about each run of a BATCH_CYCLE.
CONTROL_TEST_RUN	This is where the execution of each Control Test is logged along with the expected/actual results and a derived column indicating if the test passed or failed
JOB_BATCH_CYCLE_RUN_MSG	This table contains information on errors, exceptions and informational alerts raised or encountered during the run of a particular job (process) within a batch cycle. Note that JOB_ID 0 is reserved and has special meaning. Errors or messages raised with JOB_ID 0, refer to the entire job, not a particular process. These types of errors or messages usually mean problems were encountered initializing or setting up the BATCH_CYCLE_RUN.
BATCH_CYCLE_RUN_JOB	Holds the start and stop times for each job, or process that was executed within a particular BATCH_CYCLE_RUN.

Appendix B. ABC Batch Type Codes

ABC Batch Type Code	Batch Interval Description
00MIN	The data integration (DI) process runs continuously in real-time
30SEC	The Batch Cycle runs every 30 seconds at the 0th and 30th second of each minute
01MIN	The batch cycle runs every minute on the 0th second of the minute
05 MIN	The batch cycle runs every 5 minutes at 00, 05, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 minutes past the hour.
15 MIN	The batch cycle runs every 15 minutes at 00, 15, 30, and 45 minutes past each hour
30 MIN	Batch cycle runs twice each hour at 00, and 30 minutes past the hour.
HOUR	The batch cycle runs every hour on the hour
DAILY	The batch cycle runs every day at the hour it was scheduled
WEEK	Batch cycle runs weekly on the day and hour it was scheduled
WEEKM	Batch cycle runs weekly on Mondays on the time it was scheduled
WEEKT	Batch cycle runs weekly on Tuesdays on the time it was scheduled
WEEKW	Batch cycle runs weekly on Wednesdays on the time it was scheduled
WEEKH	Batch cycle runs weekly on Thursdays on the time it was scheduled
WEEKF	Batch cycle runs weekly on Fridays on the time it was scheduled
WEEKS	Batch cycle runs weekly on Saturdays on the time it was scheduled
WEEKU	Batch cycle runs weekly on Sundays on the time it was scheduled
MONTH	The batch cycle runs once per month on the day of the month and time at which it was scheduled
F_MON	The batch cycle runs on the first day of each month on at the time it was scheduled
M_MON	The batch cycle runs on the 15th day of each month at the time it was scheduled
L_MON	The batch cycle runs on the last day of the month on the time it was scheduled
F_QTR	The batch cycle runs on the first day of each calendar year quarter at the time it was scheduled
L_QTR	The batch cycle runs on the last day of each calendar year quarter at the time it was scheduled
FYEAR	The batch cycle runs on the first day of each calendar year at the time it was scheduled
LYEAR	The batch cycle runs on the last day of each calendar year at the time it was scheduled
FFMON	The batch cycle runs on the first day of each Fiscal month on at the time it was scheduled
LFMON	The batch cycle runs on the last day of the fiscal month on the time it was scheduled
FFQTR	The batch cycle runs on the first day of each fiscal year quarter at the time it was scheduled

LFQTR	The batch cycle runs on the last day of each fiscal year quarter at the time it was scheduled
FFYER	The batch cycle runs on the first day of each fiscal year at the time it was scheduled
LFYER	The batch cycle runs on the last day of each fiscal year at the time it was scheduled

Support:

www.eltmaestro.com/support
support@eltmaestro.com