

# AYI ACADEMY: TALEND

## Material práctico



### Módulo 1: Working with Files

## Trabajando con archivos: tratamiento de cada tipo de datos

Este módulo le enseña cómo manejar diferentes tipos de archivos estructurados y cómo realizar operaciones básicas en los archivos (leer, copiar, eliminar y enumerar). Se centra en el uso de componentes de Talend Studio dedicados a la lectura de datos de archivos estructurados (formato CSV) y archivos jerárquicos (formatos JSON y XML).

### Abriendo el proyecto

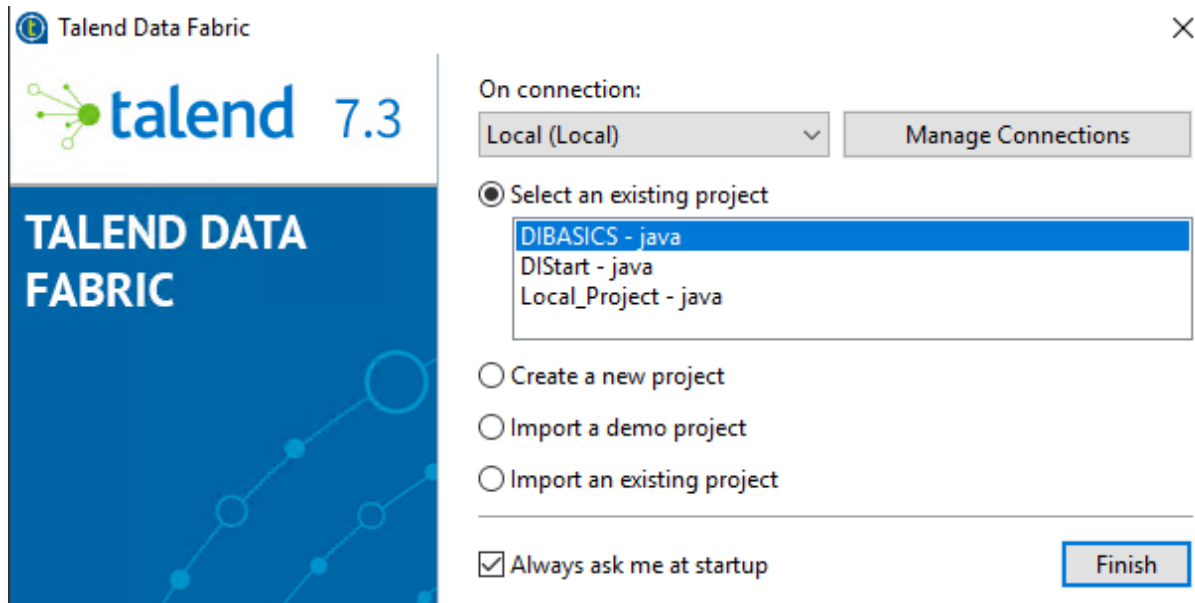
- 1) Haga clic en el icono de Studio para iniciar Talend Studio.



2) Abra el proyecto DIBASICS.

Confirme que el proyecto DIBASICS está disponible en la lista de proyectos existentes.

Si aparece el proyecto DIBASICS, selecciónelo y luego haga clic en Finalizar.



Si falta el proyecto, siga estos pasos para crearlo:

Para **On connection**, seleccione **Local (Local)**, lo que significa que cuando se crea, el proyecto se almacena localmente, en su máquina.

Haga clic en el botón de **Create a new project** e ingrese DIBASICS.



## TALEND DATA FABRIC

On connection:

Local (Local) ▾

Manage Connections

☐ Select an existing project

DStart - java

Local\_Project - java

☒ Create a new project: DIBASICS

Create

☐ Import a demo project

☐ Import an existing project

☒ Always ask me at startup

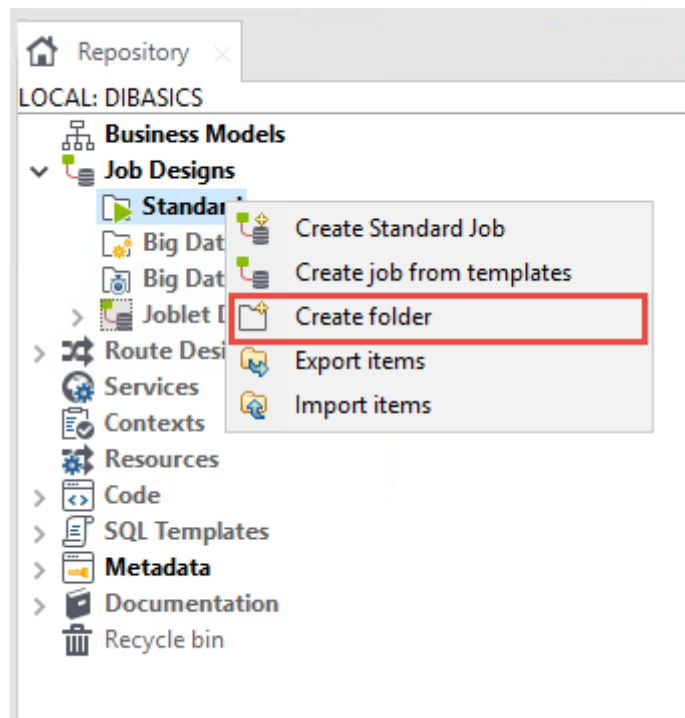
Finish

## Configura tu Proyecto

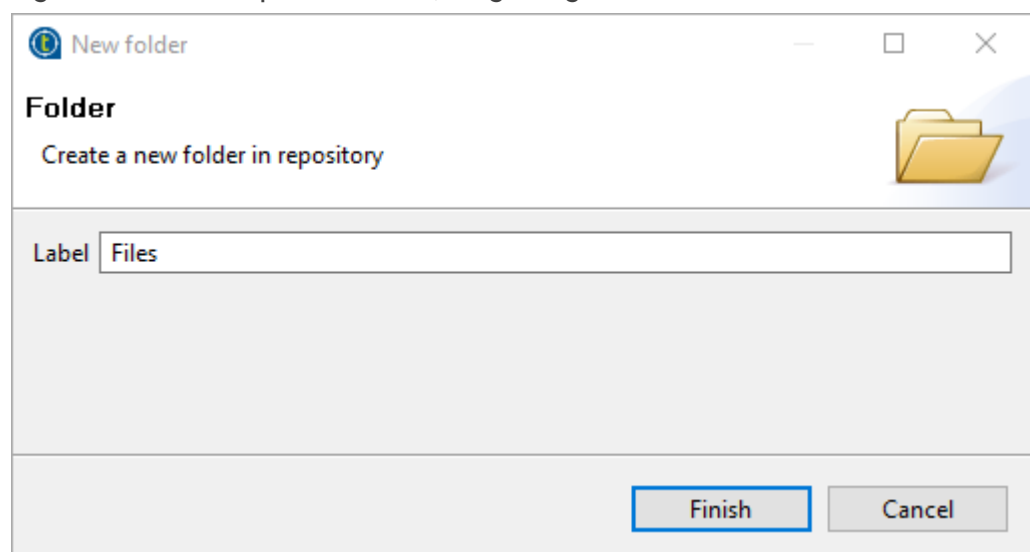
- 1) En la barra de herramientas de acceso rápido, asegúrese de que la perspectiva de **Integration** esté seleccionada.



- 2) Cree una nueva carpeta para sus jobs. En el Repositorio, expanda **Job Designs**, haga clic con el botón derecho en **Standard** y seleccione **Create folder**.



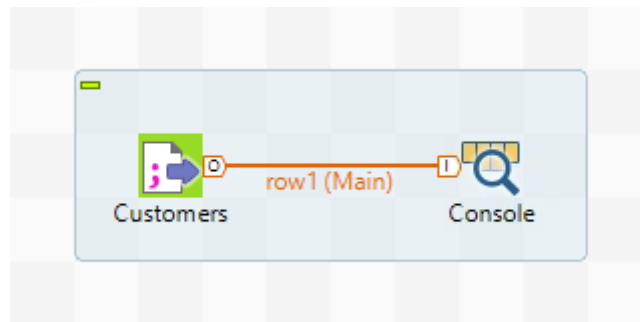
Ingresa "archivos" para la **Label**, luego haga clic en Finalizar.



La nueva carpeta aparece en **Repository > Job Designs > Standard**.

Este ejercicio le enseña cómo leer e interpretar datos provenientes de un archivo delimitado.

Al final, el Job se ve así. Lee datos de un archivo CSV y los imprime en la consola.



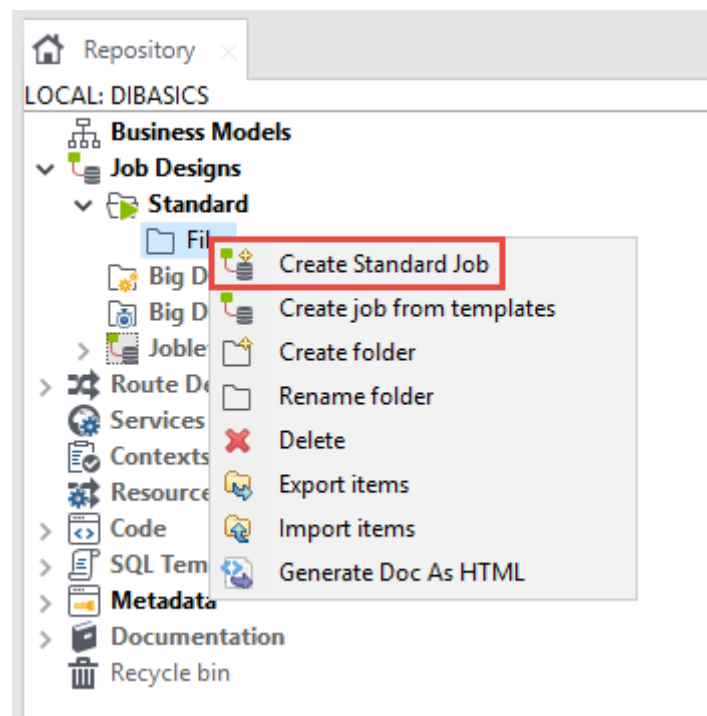
Lectura de datos de un archivo delimitado

En este ejercicio, utilizará el archivo Customers.csv ubicado en la carpeta C:\StudentFiles\DIBasics\Files como ejemplo. Examine el contenido de este archivo, como se muestra a continuación, y observe que cada línea contiene información básica del cliente, separada por comas. La primera línea es un encabezado que contiene los nombres de las columnas de datos.

```
Customers.csv
1 First, Last, Number, Street, City, State
2 Bill, Coolidge, 85013, Via Real, Austin, IL
3 Thomas, Coolidge, 63489, Lindbergh Blvd, Springfield, ca
4 Harry, Ford, 97249, Monroe Street, Salt Lake City, ca
5 Warren, McKinley, 82589, Westside Freeway, Concord, ak
6 Andrew, Taylor, 29886, Padre Boulevard, Madison, CA
7 Ulysses, Coolidge, 98646, Bayshore Freeway, Columbus, MN
8 Theodore, Clinton, 12292, San Marcos, Bismarck, NY
9 Benjamin, Jefferson, 82077, Carpinteria North, Sacramento, ca
10 William, Van Buren, 21712, Tully Road East, Albany, IL
11 Calvin, Washington, 50742, Richmond Hill, Charleston, ca
12 Jimmy, Polk, 76143, Richmond Hill, Salt Lake City, AK
13 Calvin, Adams, 52386, Lake Tahoe Blvd., Montgomery, NY
14 Ulysses, Monroe, 70511, Jones Road, Trenton, IL
15 Zachary, Tyler, 45040, Santa Rosa North, Carson City, AK
16 Ulysses, Johnson, 19989, Via Real, Juneau, AL
```

Utilice un componente tFileInputDelimited para leer un archivo delimitado fila por fila, dividir datos en campos y enviar los campos, como se define en el esquema, al siguiente componente.

- 1) Cree un nuevo trabajo en la carpeta Archivos y asígnele el nombre readDelimitedFiles.
  - a) En el **Repository**, expanda **Job Designs > Standard** y haga clic con el botón derecho en **Files**.
  - b) Seleccione **Create Standard Job**. Se abre el cuadro de diálogo New Job.



- c) En el cuadro de texto **Name**, ingrese readDelimitedFile.
    - d) Rellene los cuadros de texto **Purpose** y **Description** como se muestra.
    - e) Haga clic en Finalizar. El Trabajo se abre en el Designer.

**New Job**  
Add a job in the repository

Name: readDelimitedFile

Purpose: Import customers data

Description: Data source is a CSV file

Author: user@talend.com

Locker:

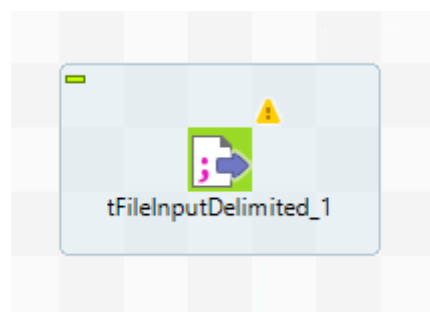
Version: 0.1 M m

Status:

Path: Files Select

Finish Cancel

- 2) Agregue un componente que lea desde un archivo delimitado.
  - a) Haga clic en cualquier lugar en el **Designer**.
  - b) Comience a escribir el nombre del componente, tFileInputDelimited. A medida que escribe, aparece una lista de componentes coincidentes.
  - c) Haga doble clic en tFileInputDelimited en la lista. El componente aparece en el Diseñador.



- 3) Cambie el nombre del componente a Clientes.
  - a) Haga clic en la pestaña **View** en la **Component** view.
  - b) Reemplace el texto en el cuadro de texto **Label format** con Clientes.  
Esto actualiza automáticamente el nombre del componente en Designer.

Job(readDelimit... Contexts(readDel... Component x Run (Job readDel...

**Customers(tFileInputDelimited\_1)**

Basic settings	Label format	Customers
Advanced settings	Hint format	<b>_UNIQUE_NAME_</b> _COMMENT_</br>
Dynamic settings	Connection format	row
<b>View</b>		
Documentation		
Validation Rules		

4) Configure la configuración básica de componentes.

Para configurar el componente Clientes, especifique las propiedades del archivo, como el nombre del archivo para leer, los separadores de filas y campos. Los datos se leen del archivo fila por fila, utilizando el separador de fila para identificar el final de una línea. Cada fila se separa a su vez en campos utilizando el separador de campos y se interpreta de acuerdo con el esquema.

- En la vista de componentes, abra la pestaña **Basic settings**.
- Para seleccionar el **File name**, haga clic en [...] y seleccione la ruta del archivo C:/StudentFiles/DIBasics/Files/Customers.csv.
- Deje el **Row Separator** predeterminado como "\n", que corresponde al final de la línea.
- Para especificar que los campos de una fila estén separados por comas, en **Field Separator**, ingrese ","
- En el cuadro de texto **Header** , ingrese 1. El componente ignora la primera línea del archivo, que no contiene datos reales, solo los nombres de las columnas de datos.

La configuración del componente se ve así:

Job(readDelimit... Contexts(readDel... Component x Run (Job readDel... Test Cases Integration Action - +

**Customers(tFileInputDelimited\_1)**

Basic settings	Property Type	Built-In	
Advanced settings	Schema	Built-In	Edit schema
Dynamic settings	"When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0."		
View	File name/Stream	C:/StudentFiles/DIBasics/Files/Customers.csv	
Documentation	Row Separator	\n	Field Separator ,
Validation Rules	CSV options		
	Header	1	Footer 0 Limit
	<input checked="" type="checkbox"/> Skip empty rows <input type="checkbox"/> Uncompress as zip file <input type="checkbox"/> Die on error		

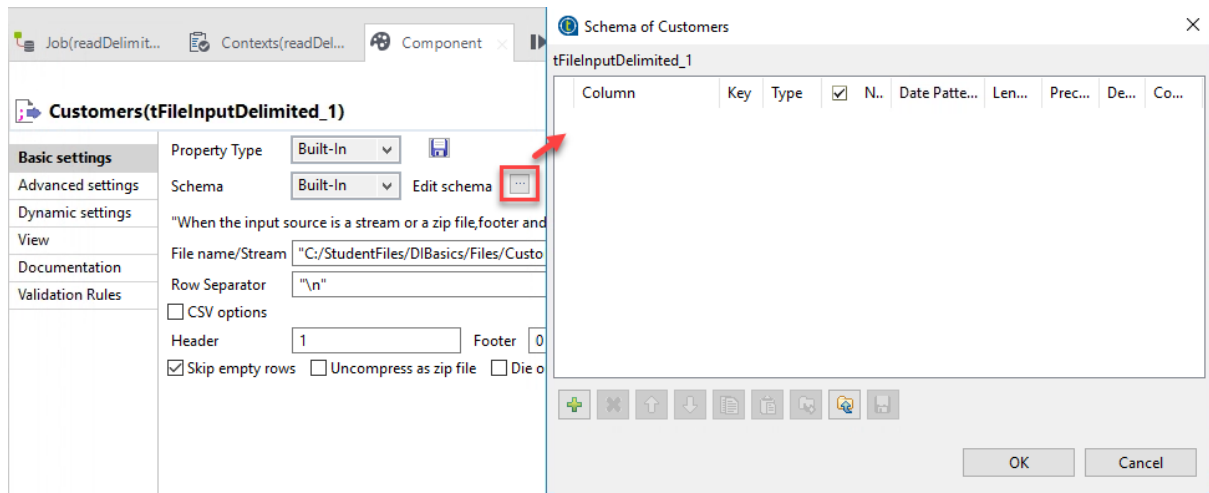
5) Definir el esquema

Por cada línea leída en el archivo, el componente asigna cada campo de datos a una columna de su esquema definido. Para cada campo de datos



extraído del archivo, el esquema necesita una columna correspondiente con un nombre y atributos, como tipo y longitud.

- a) Para definir manualmente el esquema, a la derecha de **Edit schema**, haga clic en [...]



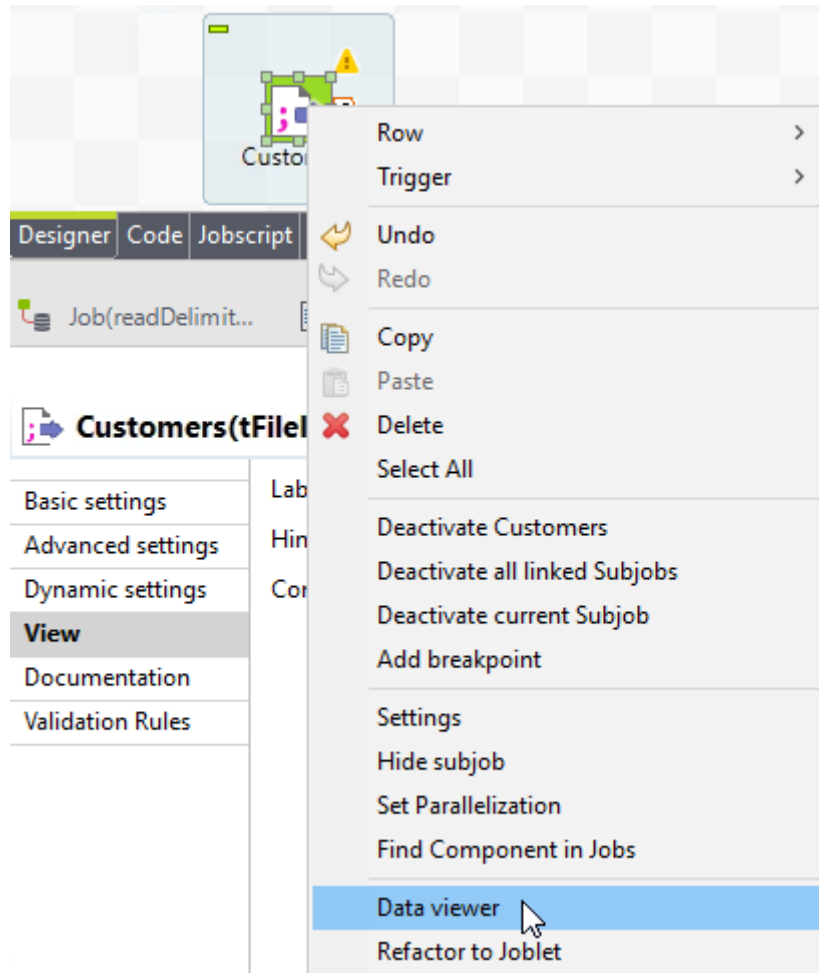
- b) Haga clic en [+] para agregar una nueva columna al esquema. Asigne el nombre **Primero** a la columna. Para **Type**, seleccione **String** y para **Length**, ingrese 15. Esto significa que la primera columna de cada fila del archivo contiene un valor de **String** con un máximo de 15 caracteres que representan el nombre del cliente.
- c) Repita estos pasos para agregar el resto de las columnas a su esquema. Seleccione los tipos y longitudes de la siguiente manera:

<i>Last</i>	<b>String</b>	15
<i>Number</i>	<b>Integer</b>	10
<i>Street</i>	<b>String</b>	20
<i>City</i>	<b>String</b>	20
<i>State</i>	<b>String</b>	2

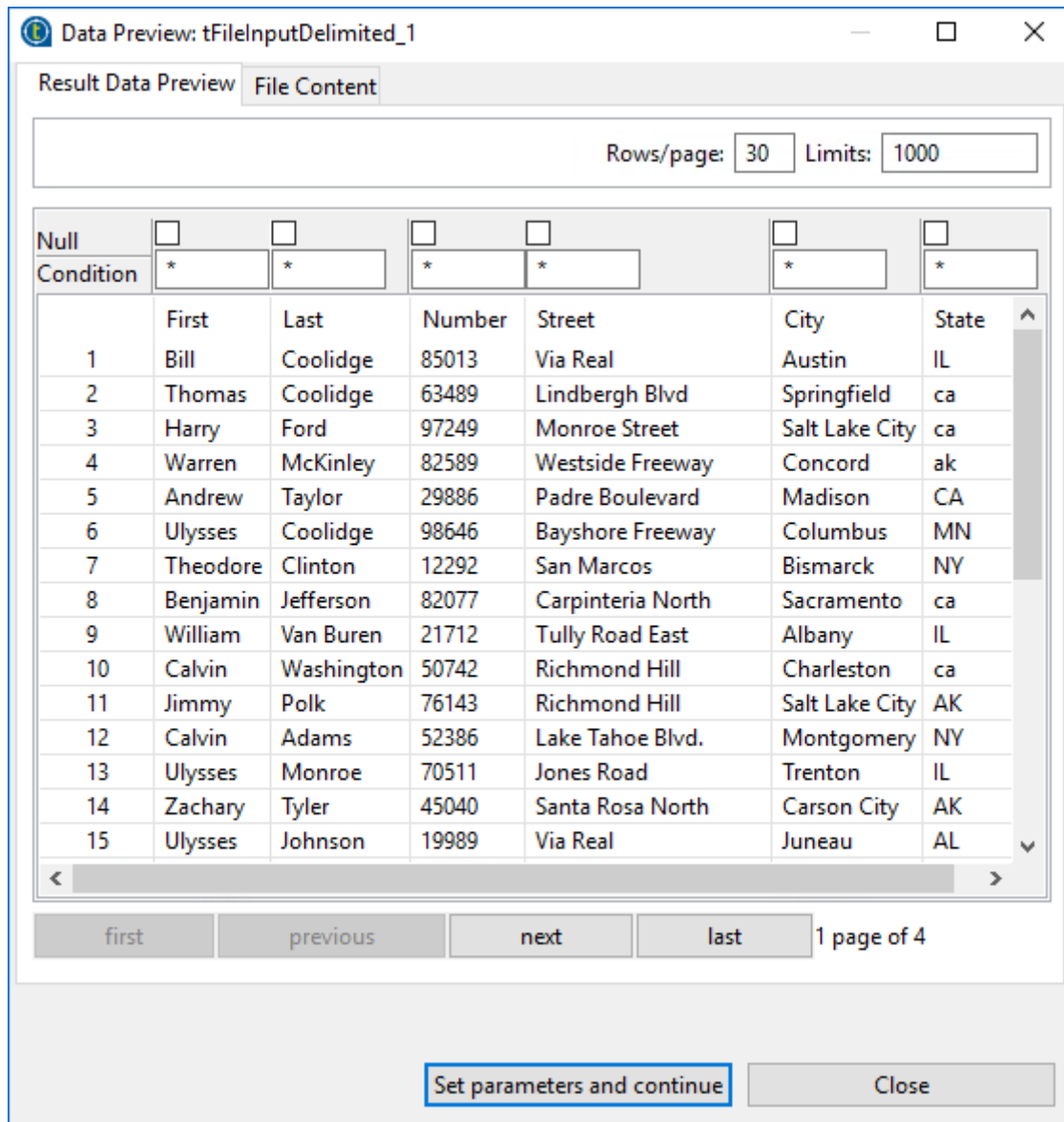
Confirme que la definición del esquema es correcta como se muestra a continuación y haga clic en Aceptar.



- 6) Utilice el visor de datos del componente Clientes para obtener una vista previa de los datos extraídos del archivo.
- a) Haga clic con el botón derecho en el componente Clientes.
  - b) Seleccionar **Data viewer**



Una ventana muestra una vista previa de los datos recuperados del archivo Customers.csv. Esta es una buena manera de confirmar que la configuración del lector de archivos se configuró correctamente. La pestaña Contenido del archivo muestra el contenido sin procesar del archivo.

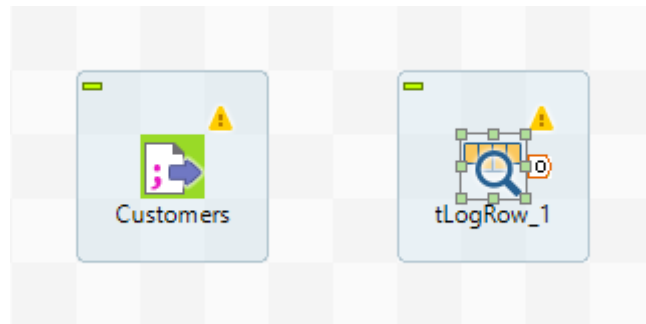


Click **Close**.

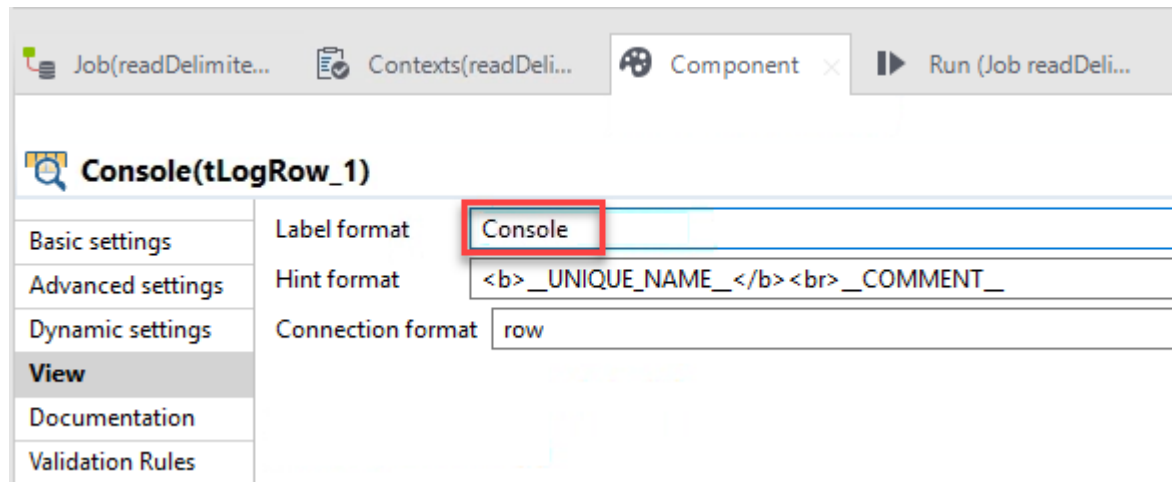
Impresión del contenido del archivo en la consola

Para mostrar el contenido leído del archivo en la consola, conecte el componente de entrada del archivo a un componente de salida tLogRow.

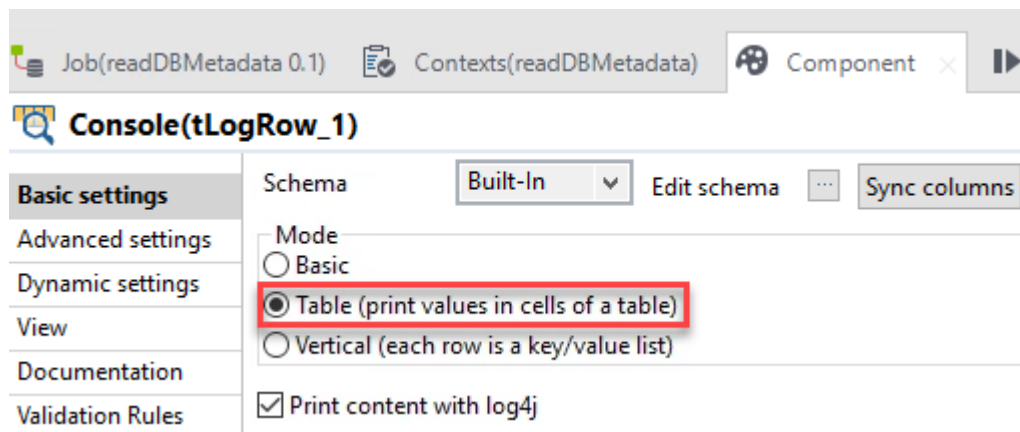
- 1) Agregue un componente tLogRow a la derecha del componente Clientes.



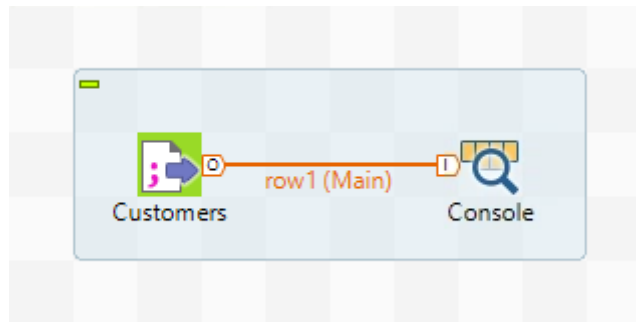
- 2) Cambie el nombre del componente tLogRow a Consola.



- 3) Configure el componente tLogRow para imprimir los resultados en una tabla.
  - a) En la vista de componentes, abra la pestaña **Basic settings**.
  - b) Para **Mode**, seleccione **Table**, que le permite imprimir los resultados en una tabla.



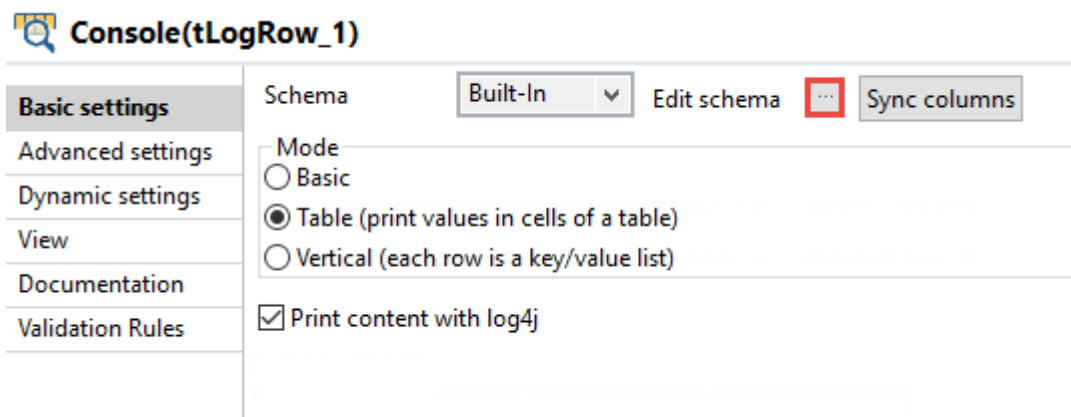
- 4) Conecte los componentes Clientes y Consola mediante una fila Principal.
  - a) Haga clic con el botón derecho en el componente de entrada, Clientes.
  - b) Select **Row** > **Main**. A connection link is visible between the input component and mouse pointer.
  - c) Haga clic en el componente **Console** para crear el enlace.



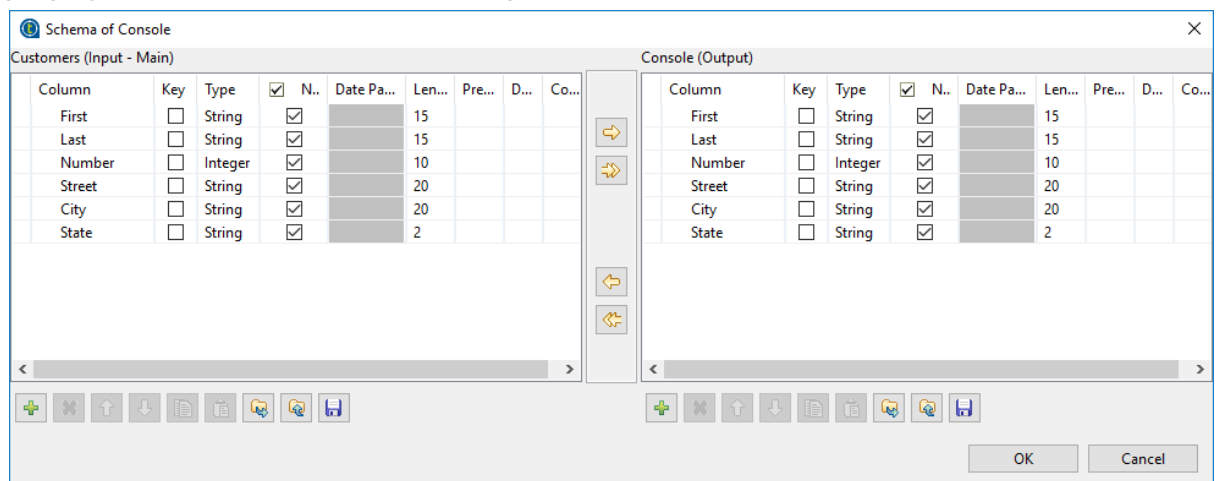
tip: Las conexiones de fila iteran en filas secuencialmente y transfieren datos entre componentes. La fila principal es el tipo de conexión de fila más común y se usa de forma predeterminada.

5) Revisa el esquema.

Seleccione el componente tLogRow y haga clic en el botón **Edit schema** ([...]) en la vista de componentes.



Observe que el esquema que especificó en el componente tFileInputDelimited se propagó cuando vinculó los dos componentes.



Click **OK**.

6) Ejecute el trabajo.

a) Abra la vista **Run** debajo del **Designer**.

b) Haga clic en **Run**. Se ejecuta el Job y se muestra en la consola la tabla que contiene la lista de clientes extraída del archivo. Confirme que el

trabajo termina con **[exit code = 0]**, lo que significa que se ejecutó con éxito.

The screenshot shows the 'Job readDelimitedFiles' window. On the left is a sidebar with 'Basic Run' selected. The main area has an 'Execution' tab with 'Run', 'Kill', and 'Clear' buttons. Below these is a table of data with 7 columns. The console output at the bottom shows '[statistics] disconnected' and 'Job readDelimitedFile ended at 02:28 11/06/2018. [exit code=0]', with the last part highlighted in a red box.

William	Buchanan	86382	Bayshore Freeway	Raleigh	ak	
James	Harding	67244	N Harrison St	Phoenix	IL	
Theodore	Wilson	18980	Harry S Truman Blvd	Hartford	ca	
Richard	Madison	55838	Monroe Street	Santa Fe	ak	
Abraham	Wilson	83959	Hutchinson Rd	Nashville	NY	
John	Johnson	64872	Erringer Road	Olympia	CA	
Rutherford	Harding	46743	Steele Lane	Boston	NY	
John	Clinton	42059	Santa Ana Freeway	Columbus	ca	
Andrew	Johnson	16064	Harbor Dr	Sacramento	IL	
Thomas	Fillmore	26120	Jean de la Fontaine	Little Rock	ca	
Chester	Taft	89505	North Preisker Lane	Cheyenne	AL	
Lyndon	Cleveland	85642	North Atherton Street	Hartford	MN	
Rutherford	Polk	99837	Santa Monica Road	Boise	CA	
Thomas	Wilson	36910	Tanger Blvd	Salem	MN	
Benjamin	Taylor	81222	North Erringer Road	Concord	MN	

[statistics] disconnected  
Job readDelimitedFile ended at 02:28 11/06/2018. [exit code=0]

Este ejercicio le enseña cómo leer e interpretar datos provenientes de archivos jerárquicos.

Al final, los dos trabajos se ven así. El primer trabajo lee datos de un archivo XML y los imprime en la consola. De manera similar, el segundo trabajo lee datos de un archivo JSON y los imprime en la consola.



En este ejercicio, utilizará los archivos ProductSales.xml y ShopSales.json ubicados en la carpeta C:\StudentFiles\DIBasics\Files.

## Lectura de datos de archivos XML jerárquicos

El primer trabajo tiene que leer datos del archivo ProductSales.xml. Mire el contenido del archivo (que se muestra a continuación) para comprender su estructura. La etiqueta raíz, como lo indica su nombre, es el elemento superior del documento XML y contiene todas las etiquetas de fila. Observe que las etiquetas de fila agrupan datos sobre ventas de productos, como el nombre de la tienda, la cantidad, el nombre del cliente y el producto.

```
<root>
  <row>
    <ShopName>Shop1</ShopName>
    <Quantity>790</Quantity>
    <CustomerName>Acme Chemical</CustomerName>
    <Product>Product 34</Product>
  </row>
  <row>
    <ShopName>Shop1</ShopName>
    <Quantity>886</Quantity>
    <CustomerName>Truck-X Repair</CustomerName>
    <Product>Product 36</Product>
  </row>
  <row>
    <ShopName>Shop1</ShopName>
    <Quantity>972</Quantity>
    <CustomerName>Lee Stairworks</CustomerName>
    <Product>Product 47</Product>
  </row>
</root>
```

Para leer datos de un archivo XML, utiliza un componente tFileInputXML.

Un componente tFileInputXML extrae los datos almacenados en el archivo XML, los asigna a los campos definidos por el esquema y los transfiere al siguiente componente.

Para seleccionar elementos y atributos en documentos XML, utilice las expresiones XPath estándar. En este ejemplo, para obtener los nodos que contienen valores, el analizador debe seleccionar, comenzando desde "raíz", los hijos de los nodos de "fila". La expresión XPath correspondiente es /root/row.

### 1) Crear un trabajo.

En el Repositorio, expanda Diseños de trabajo > Estándar, luego haga clic con el botón derecho en Archivos y cree un nuevo trabajo estándar llamado readXMLFile.

El Trabajo se abre en el Diseñador.



**New Job**  
Add a job in the repository

Name: readXMLFile

Purpose: Import products sales

Description: Data source is an XML file

Author: user@talend.com

Locker:

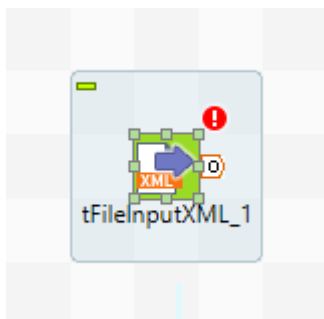
Version: 0.1 M m

Status:

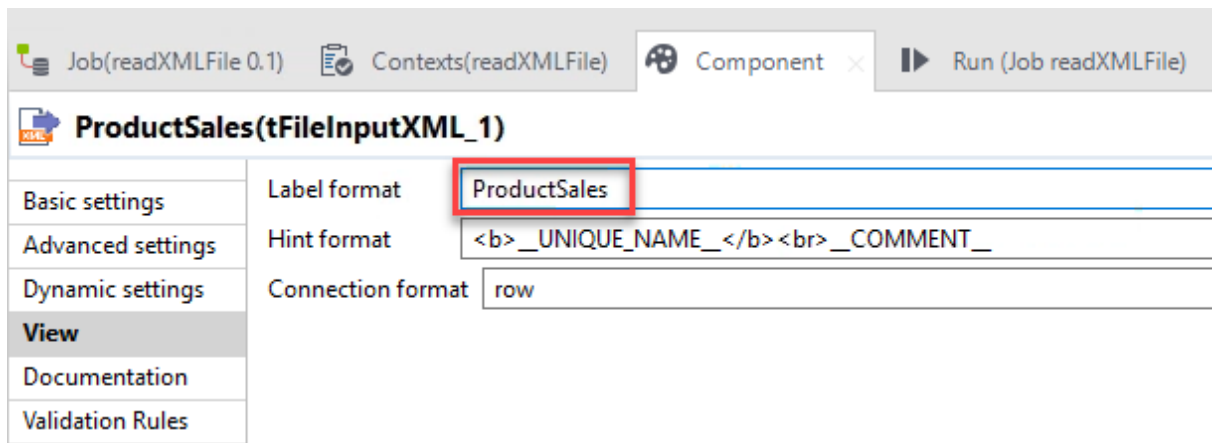
Path: Files Select

Finish Cancel

- 2) Agregue un componente que lea archivos XML  
Agregue un nuevo componente tFileInputXML al Diseñador.

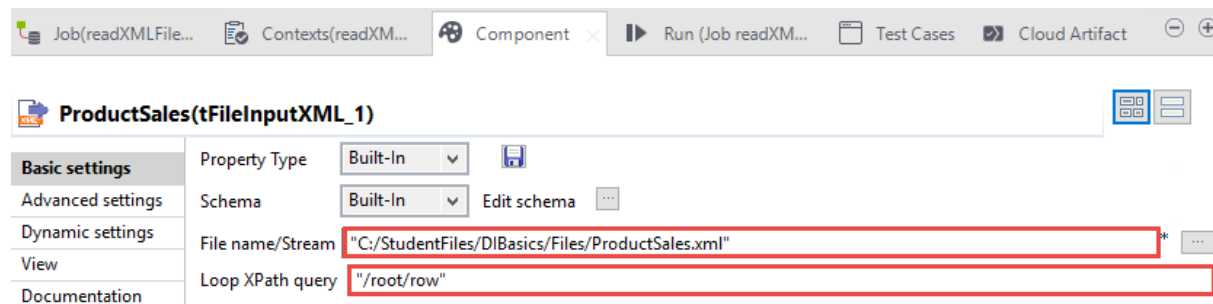


- 3) Cambie el nombre del componente tFileInputXML a ProductSales.



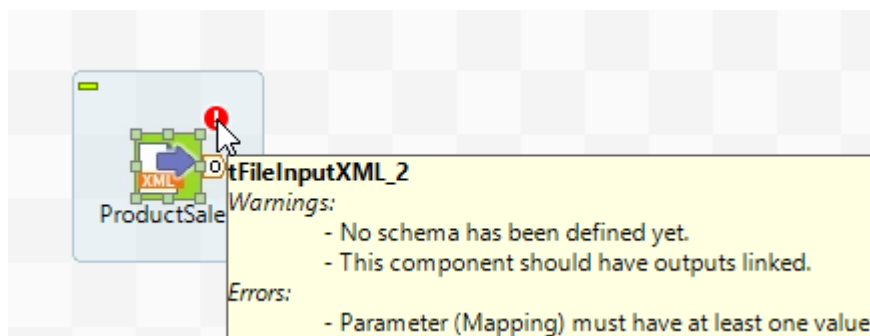
4) Configure el componente ProductSales, incluido el nombre del archivo para leer y sus propiedades.

- En la vista de componentes, abra la pestaña Configuración básica.
- Para configurar el nombre del archivo, haga clic en [...] y seleccione la ruta del archivo C:/StudentFiles/DIBasics/Files/ProductSales.xml.
- Ingresa la expresión XPath "/root/row" en el cuadro de consulta Loop XPath para seleccionar los nodos que contienen valores.



5) Inspeccionar advertencias y errores.

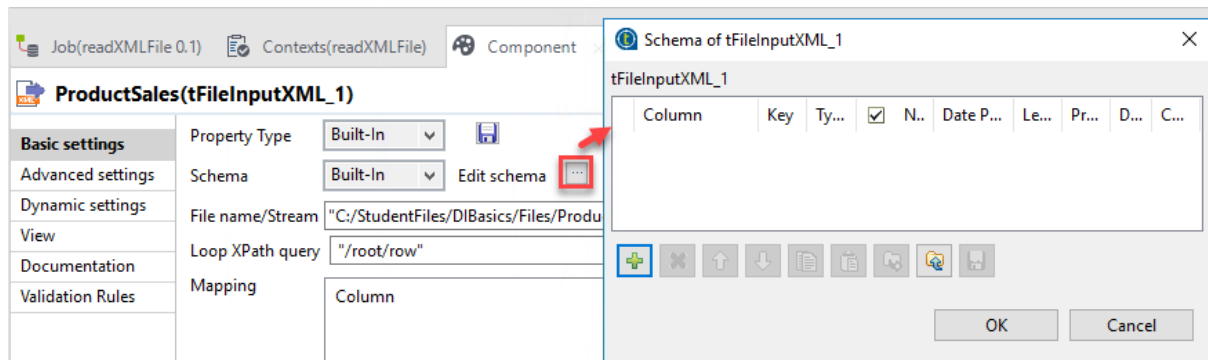
En Designer, observe que el componente ProductSales muestra un error (un icono de signo de exclamación) en la esquina superior derecha. Pase el mouse sobre el icono para mostrar el mensaje.



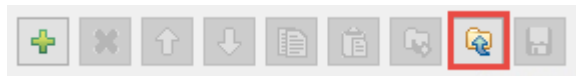
Un mensaje de advertencia muestra que el esquema no se ha definido. Además, un mensaje de error indica que no se ha realizado la asignación. Esto se debe a que el parámetro Mapeo no está configurado. Este parámetro asocia los campos de datos leídos del archivo con las columnas definidas en el esquema.

6) Cree las columnas del esquema y sus propiedades.

- a) A la derecha de Editar esquema, haga clic en [...]. Se abre un cuadro de diálogo que le permite definir un esquema para el componente.

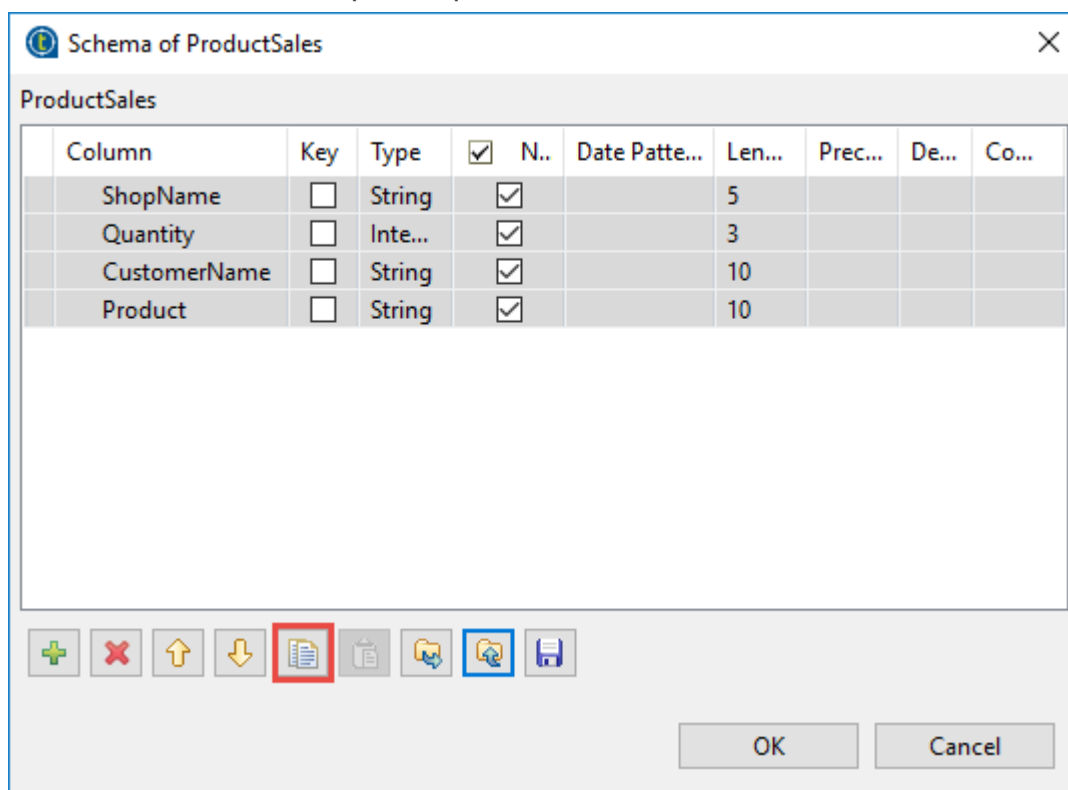


- b) Esta vez, carga el esquema desde un archivo local.



En la barra de herramientas, haga clic en la carpeta con el botón de flecha, como se muestra en la captura de pantalla, y seleccione "C:\StudentFiles\DIBasics\Files\Schema.xml".

- c) Asegúrese de que todas las columnas estén seleccionadas y haga clic en el botón Copiar esquema.



Lo reutiliza más tarde mientras configura el segundo trabajo.

- d) Para cerrar la ventana del esquema, haga clic en Aceptar.

En la vista de componentes, en la tabla de asignación, observe que la información de la columna se agrega automáticamente después de definir el esquema.

- 7) Para configurar la asignación entre los nombres de columna y el contenido real para extraer del archivo, ingrese los nombres de nodo correspondientes exactos utilizados por la estructura xml, entre comillas dobles, como se muestra.

Column	XPath query	Get Nodes
ShopName	"ShopName"	<input type="checkbox"/>
Quantity	"Quantity"	<input type="checkbox"/>
CustomerName	"CustomerName"	<input type="checkbox"/>
Product	"Product"	<input type="checkbox"/>

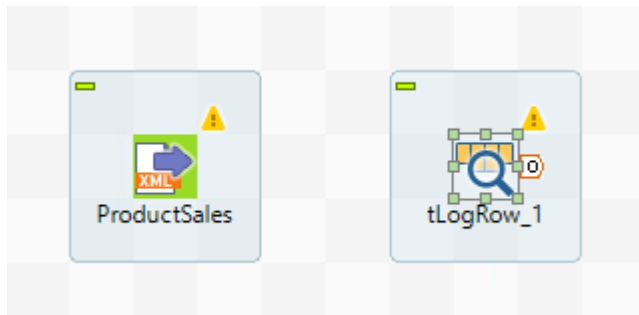
- 8) Abra el visor de datos en el componente ProductSales haciendo clic con el botón derecho en el componente y luego seleccionando Visor de datos.

Null Condition	ShopName	Quantity	CustomerName	Product
<input type="checkbox"/>	Shop1	415	Johnson, Erico & Co CPA's	Product 29
<input type="checkbox"/>	Shop1	231	Gordon & Son Brokerage Ltd.	Product 47
<input type="checkbox"/>	Shop1	790	Acme Chemical Supply	Product 34
<input type="checkbox"/>	Shop1	886	Truck-X Muffler Repair	Product 36
<input type="checkbox"/>	Shop1	783	Truck-X Muffler Repair	Product 45
<input type="checkbox"/>	Shop1	972	Lee Stairworks and Carpentry,	Product 47
<input type="checkbox"/>	Shop1	884	Lambert & Locke Legal Services	Product 45
<input type="checkbox"/>	Shop1	730	Gourmet the Frog	Product 44
<input type="checkbox"/>	Shop1	692	Elm Place Business Supplies	Product 48
<input type="checkbox"/>	Shop1	404	Truck-X Muffler Repair	Product 3
<input type="checkbox"/>	Shop1	230	George Lee Antiques	Product 23
<input type="checkbox"/>	Shop1	834	DBN Bank	Product 4
<input type="checkbox"/>	Shop1	668	Pivot Point College	Product 20
<input type="checkbox"/>	Shop1	328	Arthur M. Feildberg Galleries	Product 24
<input type="checkbox"/>	Shop1	366	Cradle the Call	Product 30
<input type="checkbox"/>	Shop1	409	Johnson, Erico & Co CPA's	Product 31

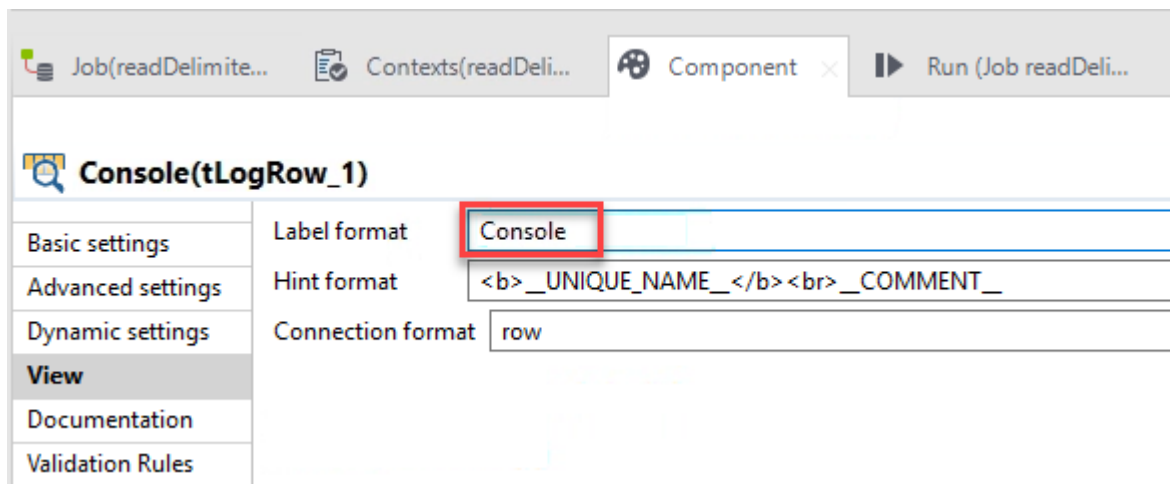
Impresión del contenido del archivo en la consola

Para imprimir el contenido leído del archivo en la consola, conecte el lector de archivos a un componente tLogRow.

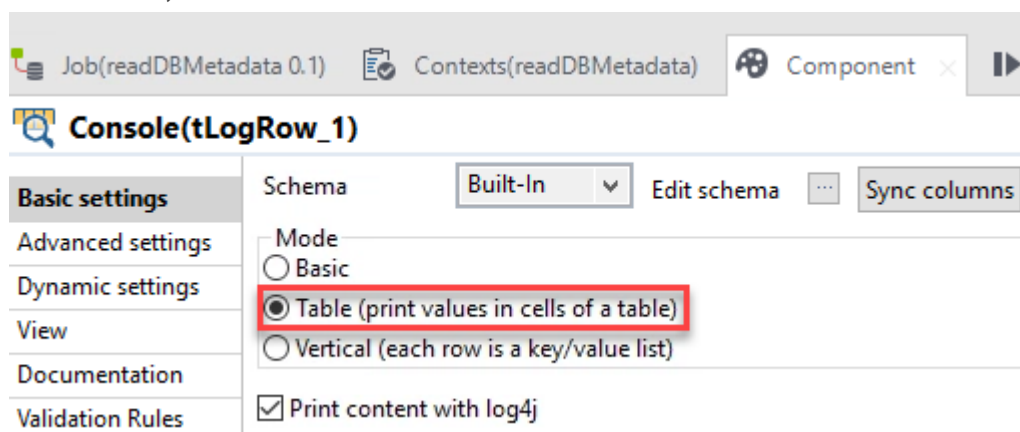
- 1) Agregue un nuevo componente tLogRow a la derecha del componente ProductSales.



- 2) Cambie el nombre del componente tLogRow a Console.

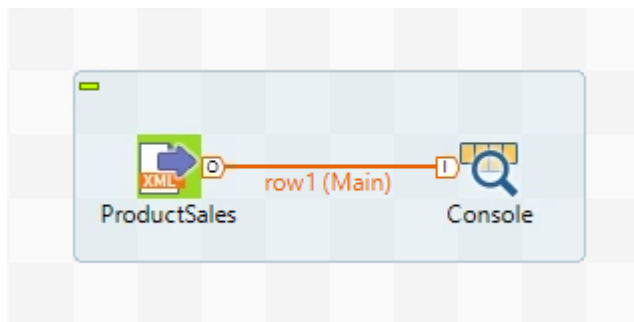


- 3) Configure el componente tLogRow para imprimir los resultados en una tabla.
  - a) En la vista de componentes, seleccione la pestaña Configuración básica.
  - b) Para Modo, seleccione Tabla.



- 4) Conecte los componentes ProductSales y Console.

Haga clic con el botón derecho en el componente de entrada, ProductSales, y use una fila Principal para conectarlo a la Consola.



5) En la vista Ejecutar, haga clic en Ejecutar.

Job(readX... Contexts(re... Component Run (Job re... Test Cases Integration...

### Job readXMLFile

Basic Run  
Debug Run  
Advanced settings  
Target Exec  
Memory Run

Execution

Run Kill Clear

Shop3	662	Cotton Thumb Ltd.	Product	31
Shop3	817	Car Suite Cleaners	Product	3
Shop3	412	Garfield Appliance Service	Product	5
Shop3	376	Gordon & Son Brokerage Ltd.	Product	36
Shop3	47	Rythmics Ltd.	Product	37
Shop3	501	Glenn Oaks Office Supplies	Product	44
Shop3	909	Resnick DDS	Product	27
Shop3	826	Eva Galler Galleries Inc.	Product	30
Shop3	211	Crossroads Construction, Inc.	Product	47
Shop3	22	Gordon & Son Brokerage Ltd.	Product	30
Shop3	156	Toggle & Myerson Ltd	Product	34
Shop3	752	PGS Associates	Product	49
Shop3	753	Deer Lake Inn	Product	6

[statistics] disconnected

Job readXMLFile ended at 01:22 27/08/2018. [exit code=0]

El trabajo se ejecuta y la tabla que contiene los elementos extraídos del archivo se muestra en la consola. Confirme que el trabajo termina con [código de salida = 0], lo que significa que se ejecutó con éxito.

En el siguiente ejercicio, creará un trabajo similar que lee datos de un archivo JSON.

### Lectura de datos de archivos JSON jerárquicos

Examine el archivo ShopSales.json, que se muestra a continuación, para comprender su estructura. Tenga en cuenta que contiene el mismo tipo de información que el archivo XML anterior, solo que el formato es diferente. Ubique los campos que le interesan: nombre de la tienda, cantidad, producto y nombre del cliente.

```

{
  "ShopSales":
  [
    {
      "ShopName": "Shop1",
      "Quantity": 415,
      "Product": "Product 29",
      "CustomerName": "Johnson, Erico & Co CPA's"
    },
    {
      "ShopName": "Shop1",
      "Quantity": 231,
      "Product": "Product 47",
      "CustomerName": "Gordon & Son Brockerage Ltd."
    },
    {
      "ShopName": "Shop1",
      "Quantity": 790,
      "Product": "Product 34",
      "CustomerName": "Acme Chemical Supply"
    }
  ]
}

```

Para leer datos de un archivo JSON, utiliza un componente tFileInputJSON.

Este componente extrae los datos almacenados en el archivo JSON y los envía a los campos definidos por el esquema, al siguiente componente.

Para este ejercicio, se utiliza el analizador JSONPath. Para configurar el analizador JSONPath, use expresiones JSONPath. JSONPath se comporta de manera muy similar a la forma en que XPath maneja documentos XML. En este ejemplo, para extraer datos sobre ventas, el analizador tiene que seleccionar, desde la parte superior del documento, todos los objetos que son hijos de ShopSales. La expresión JSONPath correspondiente es \$.ShopSales[\*]

#### 1) Crear un trabajo.

En el Repositorio, expanda Diseños de trabajo > Estándar, luego haga clic con el botón derecho en Archivos y cree un nuevo trabajo llamado readJSONFile.

**New Job**  
Add a job in the repository

Name: readJSONFile

Purpose: Import products sales

Description: Data source is a JSON file

Author: user@talend.com

Locker:

Version: 0.1 M m

Status:

Path: Files Select

Finish Cancel

- 2) Utilice el mismo método que utilizó anteriormente en este ejercicio para agregar un componente tFileInputJSON en Designer.



- 3) Cambie el nombre del componente tFileInputJSON a ShopSales.

Job(readJSONFile 0.1) Contexts(readJSONFile) Component × Run (Job readJSONFile)

**ShopSales(tFileInputJSON\_1)**

Basic settings

Advanced settings

Dynamic settings

**View**

Documentation

Validation Rules

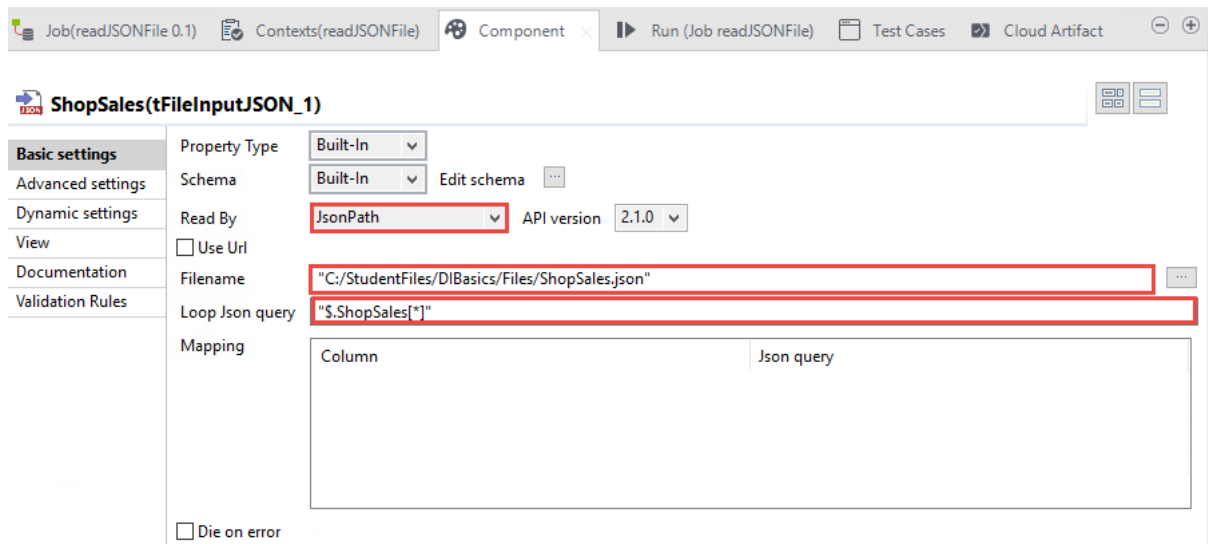
Label format: ShopSales

Hint format: **<b>\_UNIQUE\_NAME\_</b><br>\_COMMENT\_**

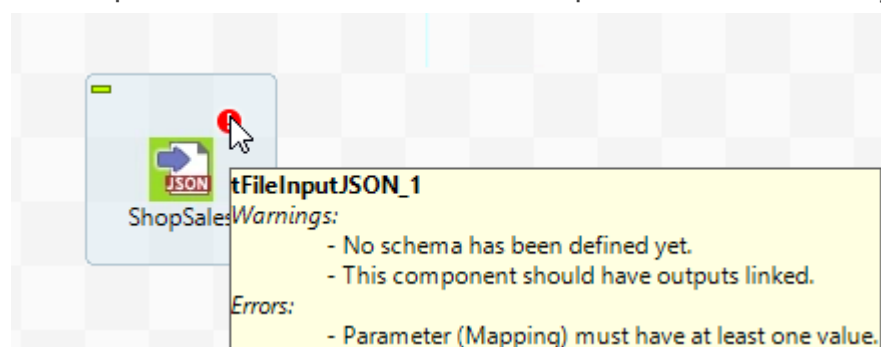
Connection format: row



- 4) Configure el componente ShopSales, incluida la ruta del archivo y las opciones del analizador.
- Haga clic en ShopSales para abrir la vista de componentes.
  - Para Leer por, seleccione JsonPath y deje la versión de API predeterminada.
  - Para seleccionar el nombre del archivo, haga clic en [...] y seleccione la ruta del archivo  
C:/StudentFiles/DIBasics/Files/ShopSales.json.
  - Para especificar cómo iterar en el documento para extraer datos, para la consulta Loop Json, ingrese la expresión JSONPath "\$.ShopSales[\*]".
- Aunque las comillas dobles no forman parte de la expresión, son necesarias porque Studio espera que esta propiedad sea una cadena Java.

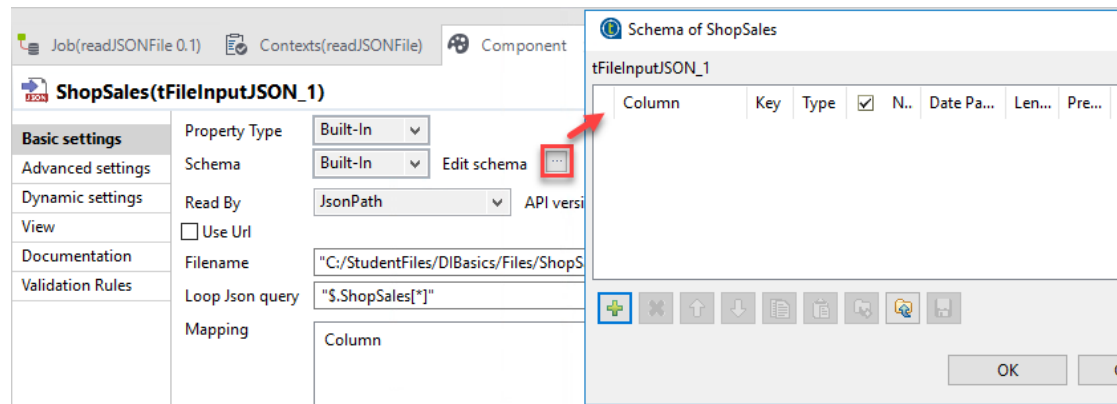


- e) En Designer, observe que el componente muestra un error (un icono de signo de exclamación) en la esquina superior derecha. Pase el puntero del mouse sobre el icono para mostrar el mensaje.

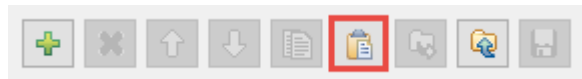


La información sobre herramientas señala que el parámetro Asignación debe tener al menos un valor. Este parámetro indica el mapeo entre los datos leídos del archivo (usando la consulta JSON) y las columnas definidas en el esquema. Para configurar la asignación, debe definir un esquema.

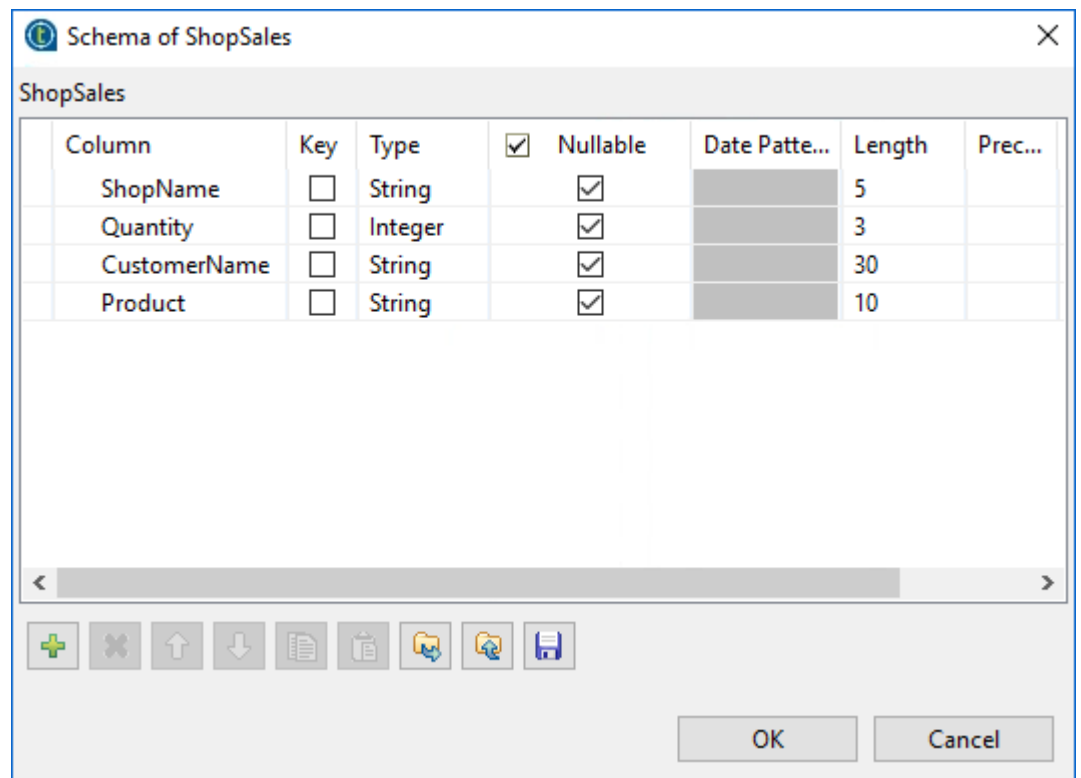
- 5) Cree las columnas del esquema y sus propiedades.
- a) A la derecha de Editar esquema, haga clic en [...]. Se abre un cuadro de diálogo que le permite definir un esquema para el componente.



- b) Para pegar el esquema copiado anteriormente, en la barra de herramientas, haga clic en el botón Pegar.



- c) Confirme que su esquema esté definido como se muestra a continuación, luego haga clic en Aceptar.



En la sección Asignación, observe que la información de la columna se agrega automáticamente después de definir el esquema.

- 6) Para configurar la asignación entre los nombres de las columnas y los campos de datos extraídos del archivo, ingrese los nombres de los

campos que se muestran en la siguiente captura de pantalla entre comillas dobles.

Job(readJSONFile 0.1) Contexts(readJSONFile) Component Run (Job readJSONFile) Test Cases Integration Action

**ShopSales(tFileInputJSON\_1)**

**Basic settings**

Property Type: Built-In  
Schema: Built-In Edit schema  
Read By: JsonPath API version: 2.1.0  
☐ Use Url  
Filename: "C:/StudentFiles/DI Basics/Files/ShopSales.json"  
Loop Json query: "\$.ShopSales[\*]"  
Mapping:

Column	Json query
ShopName	"ShopName"
Quantity	"Quantity"
CustomerName	"CustomerName"
Product	"Product"

7) Confirme la configuración de asignación abriendo el visor de datos en el componente ShopSales.

**Data Preview: tFileInputJSON\_1**

Result Data Preview File Content

Rows/page: 30 Limits: 1000

Null Condition

	ShopName	Quantity	CustomerName	Product
1	Shop1	415	Johnson, Erico & Co CPA's	Product 29
2	Shop1	231	Gordon & Son Brockering Ltd.	Product 47
3	Shop1	790	Acme Chemical Supply	Product 34
4	Shop1	886	Truck-X Muffler Repair	Product 36
5	Shop1	783	Truck-X Muffler Repair	Product 45
6	Shop1	972	Lee Stairworks and Carpentry,	Product 47
7	Shop1	884	Lambert & Locke Legal Services	Product 45
8	Shop1	730	Gourmet the Frog	Product 44
9	Shop1	692	Elm Place Business Supplies	Product 48
10	Shop1	404	Truck-X Muffler Repair	Product 3
11	Shop1	230	George Lee Antiques	Product 23
12	Shop1	834	DBN Bank	Product 4
13	Shop1	668	Pivot Point College	Product 20
14	Shop1	328	Arthur M. Feildberg Galleries	Product 24
15	Shop1	366	Cradle the Call	Product 30
16	Shop1	409	Johnson, Erico & Co CPA's	Product 31

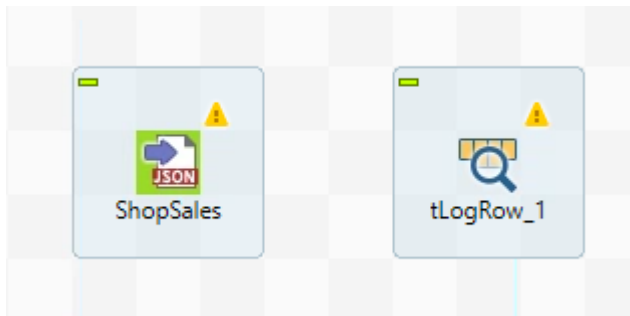
first previous next last 1 page of 17

Set parameters and continue Close

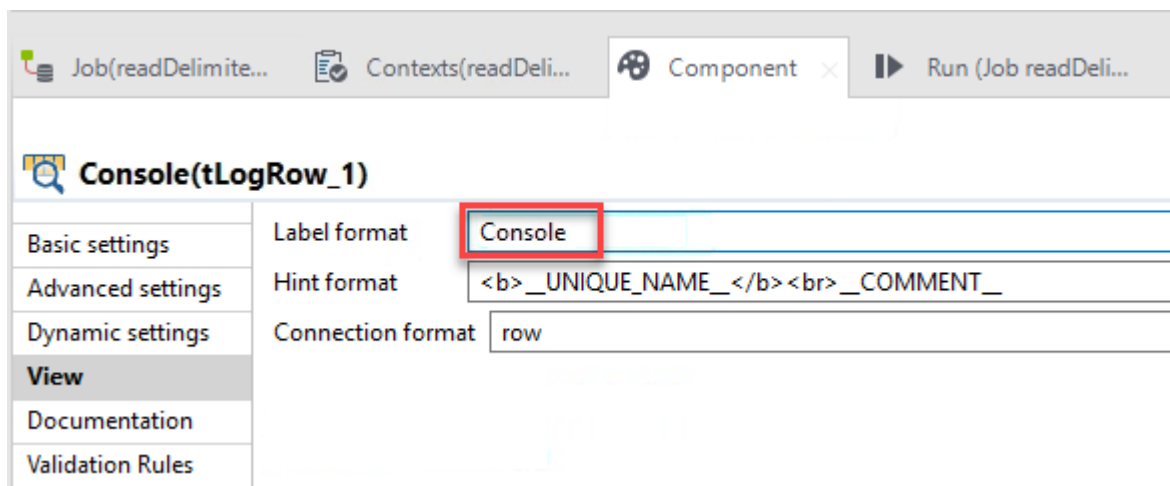
## Impresión del contenido del archivo en la consola

Para imprimir el contenido leído del archivo en la consola, conecte el lector de archivos a una consola.

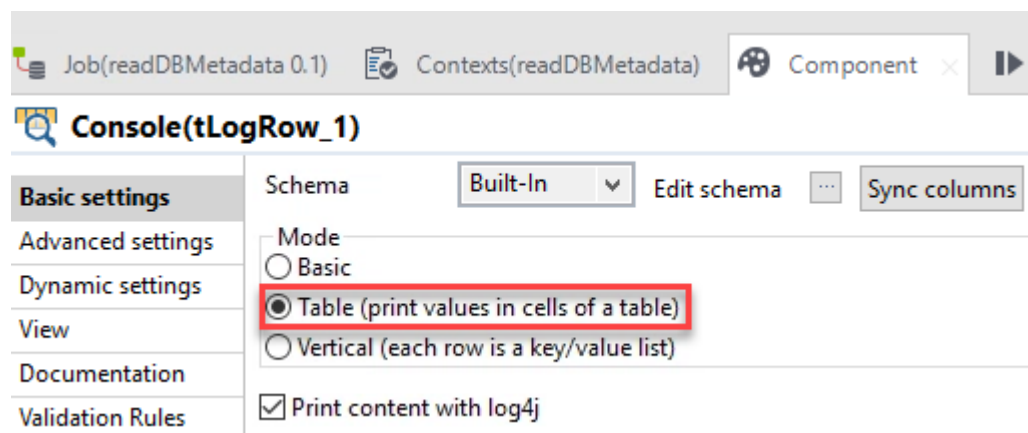
- 1) En Designer, agregue un nuevo componente tLogRow a la derecha del componente ShopSales.



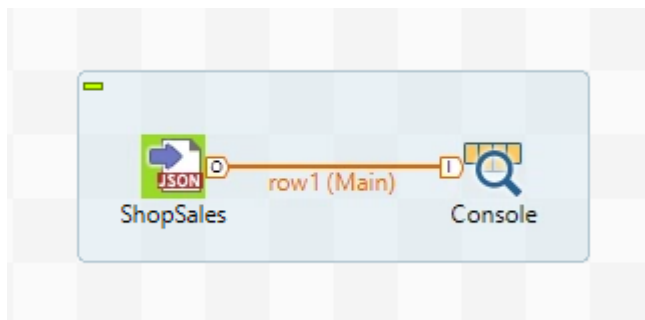
- 2) Cambie el nombre del componente tLogRow a Console.



- 3) Configure el componente tLogRow para imprimir los resultados en una tabla.
  - a) Haga clic en Consola para abrir la vista de componentes.
  - b) Para Modo, seleccione Tabla.



- 4) Conecte ShopSales y el componente Consola mediante una fila Principal.



- 5) Abra la vista Ejecutar y haga clic en Ejecutar.

Job(readJSONFile 0.1) Contexts(readJSONFile) Component Run (Job readJSONFile) Test Cases Integrati

### Job readJSONFile

**Basic Run**  
Debug Run  
Advanced settings  
Target Exec  
Memory Run

Execution

Shop1	352	Yale Remins Temp Agency	Product 44
Shop1	761	Highland Heights Gift Shop	Product 23
Shop1	896	Acturial Enterprises Ltd.	Product 14
Shop1	500	Resnick DDS	Product 15
Shop1	124	Bill's Dive Shop	Product 11
Shop1	965	George Lee Antiques	Product 39
Shop1	518	Yale Remins Temp Agency	Product 7
Shop1	26	Truck-X Muffler Repair	Product 25
Shop1	783	PGS Associates	Product 44
Shop1	630	Resnick DDS	Product 35
Shop1	682	Lee Stairworks and Carpentry.	Product 28
Shop1	912	Gourmet the Frog	Product 42
Shop1	179	Park District Of America	Product 50
Shop1	400	Quad City Cleaning	Product 42
Shop1	940	Air Mexico	Product 10

[statistics] disconnected

Job readJSONFile ended at 05:48 12/06/2018. **[exit code=0]**

☐ Line limit  ☐ Wrap

El trabajo se ejecuta y la tabla que contiene los elementos extraídos del archivo se muestra en la consola. Confirme que el trabajo termina con [código de salida = 0].