

# AYI ACADEMY: TALEND

## Material práctico



### Módulo 4: Trabajando con variables de contexto

# Módulo 4: Trabajando con variables de contexto

- **Temas:**
- **Configuración del entorno.**
- **Creación de una variable de contexto build-in.**
- **Conexión a bases de datos usando variables de contexto.**
- **Creación de un Context Group en el repositorio.**
- **Carga de variables de contexto desde un flujo (flow).**
- **Resumen.**

## Objetivos

---

Después de completar este módulo, podrá:

- Crear variables de contexto built-in específicas para un Job
- Utilizar variables de contexto en la configuración de los componentes de Talend
- Ejecutar un Job usando variables de contexto
- Conectar un Job a una base de datos usando variables de contexto
- Crear un repositorio de variables de contexto que esté disponible para todos los Jobs
- Cargar variables de contexto desde un flujo

## Caso de uso

---

Estás desarrollando Talend Jobs para una software company. En el entorno de desarrollo, se crean y prueban Jobs para que los administradores los ejecuten, posteriormente, en el entorno de producción.

Los Jobs de producción deben estar conectados a una versión productiva de la base de datos, **production**.

En el entorno de desarrollo, los Jobs usan una versión de prueba de la base de datos, **training**, la cual es una réplica de la base de datos productiva.

## Tema 1: Configuración del entorno

### Antes de comenzar

Antes de desarrollar un proyecto en Talend Studio, necesitas ejecutar Talend Studio y abrir un proyecto. Luego debes crear la carpeta contenedora de los Jobs y cargar el archivo para conectar con la base.

### Abrir un proyecto

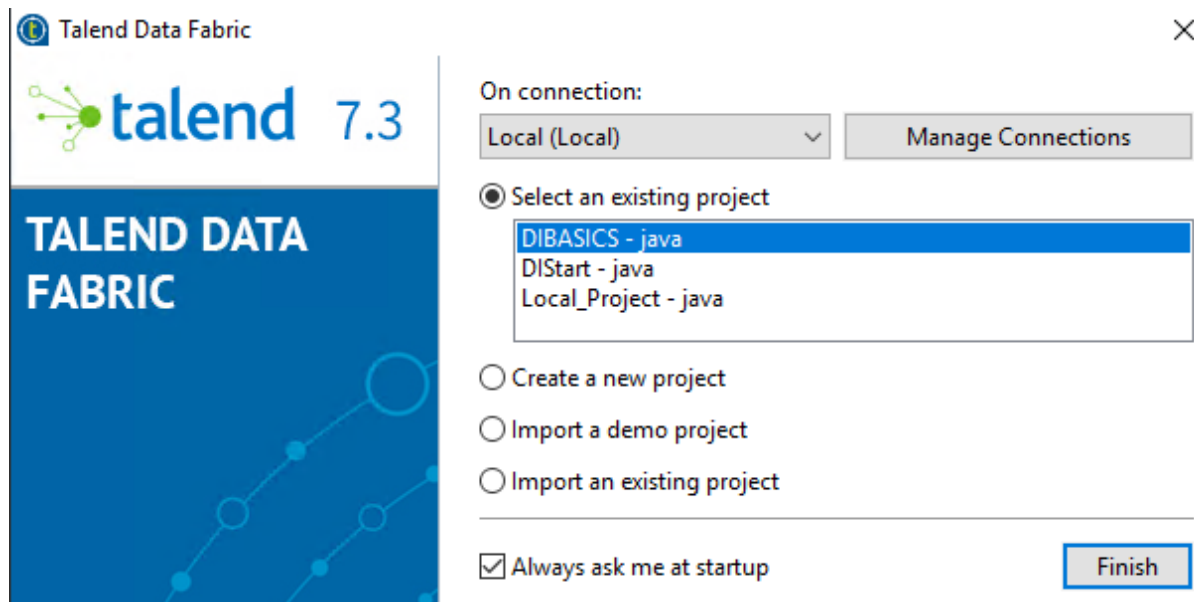
1.- Haga doble click en el ícono **Studio** para ejecutar Talend Studio.

Se muestra la pantalla de presentación de Talend Data Fabric, que le permite abrir un proyecto existente o crear uno nuevo.

2.- Abra el proyecto DIBASICS.

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

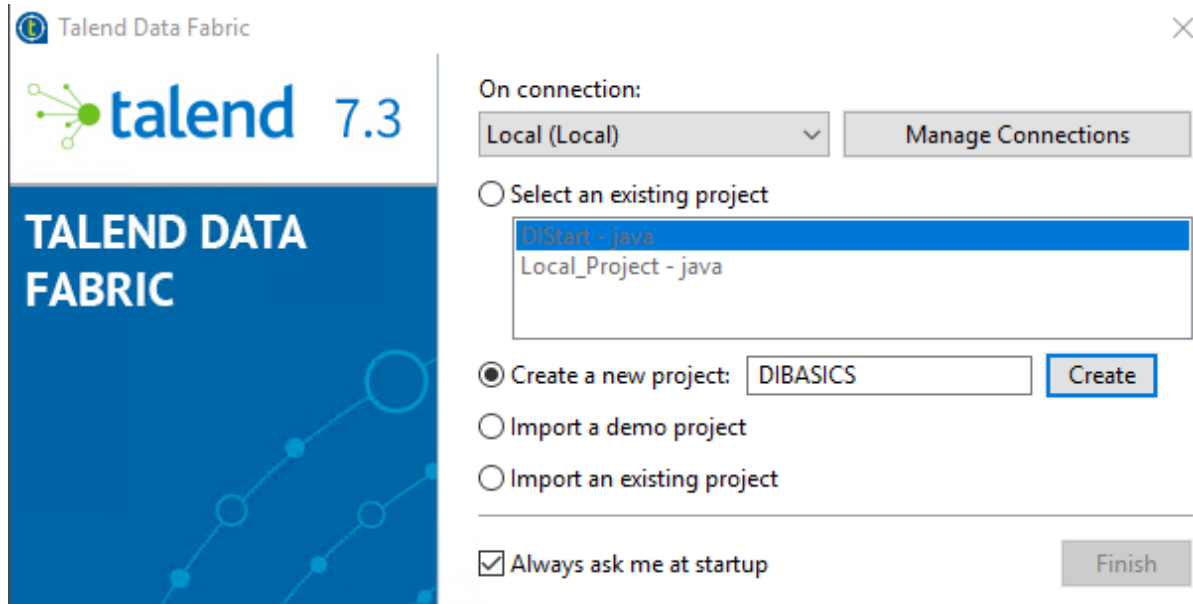
→ Si aparece el proyecto **DIBASICS**, haga clic en él y luego haga clic en **Finalizar**.



→ Si no encuentra el proyecto, siga estos pasos para crearlo:

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

b.- Haga click en **Create a new project** e ingrese *DIBASICS*.



c.- Haga click en **Create** y espere a que el proyecto aparezca en la lista. El proyecto que creaste estará seleccionado por defecto.

d.- Haga click en **Finish**.

- 💡 Si se le solicita que se conecte a Talend Community, haga clic en **Skip this Step**.
- 💡 Si se abre la página **Welcome**, haga click en **Start now!**

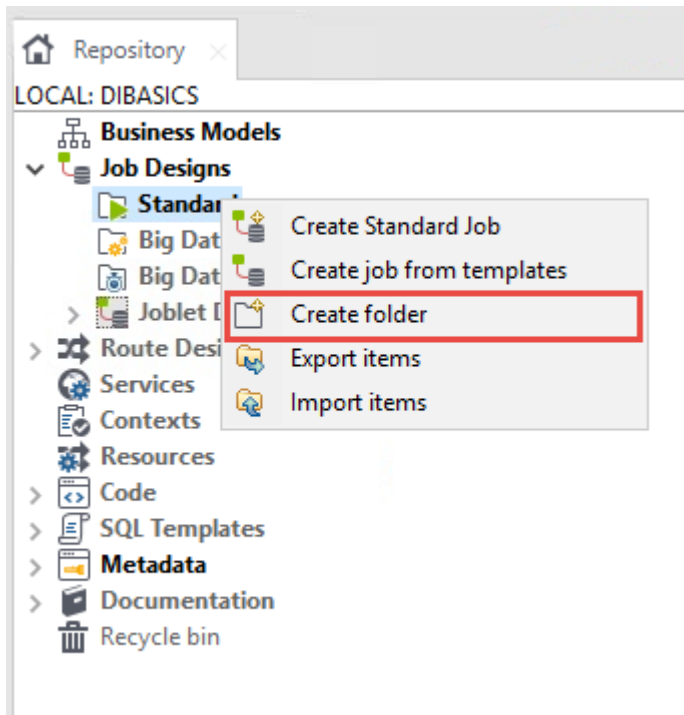
## Configurar el proyecto

1.- En la barra **Quick Access**, asegúrese que la perspectiva **Integration** esté seleccionada.

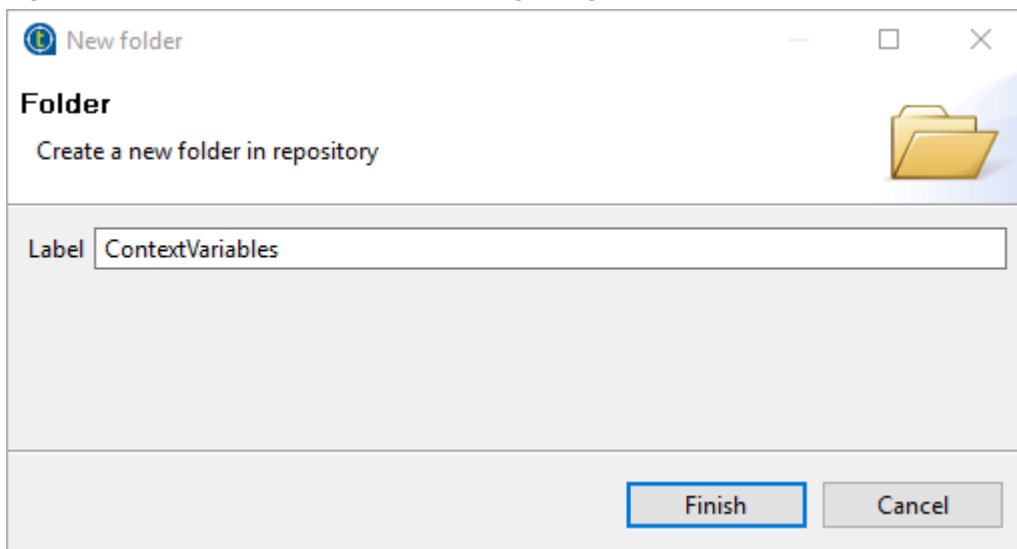


2.- Cree una carpeta para los Jobs.

En **Repository** expanda **Job Designs**, haga click derecho en **Standard**, luego seleccione **Create folder**.



Ingrese *ContextVariables* en **Label**, luego haga click en **Finish**.



La nueva carpeta se va a crear en Repository > Job Designs > Standard.

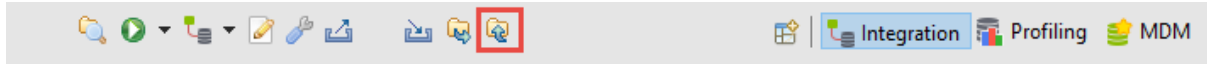
💡 Almacenar los Jobs en una carpeta lo ayudará a mantenerse organizado. Para este ejercicio, la carpeta *ContextVariables* va a ser la carpeta por defecto para almacenar los Jobs de este ejercicio.

### 3.- Cargue un archivo.

El zip *ImportContextVariables.zip* se encuentra en la carpeta **C:\StudentFiles\DI Basics\ContextVariables**.

Cargue este archivo para importar la metadata necesaria para la conexión a la base de datos sandbox.

a.- En la barra **Quick Access**, haga click en el ícono **folder and up arrow**.

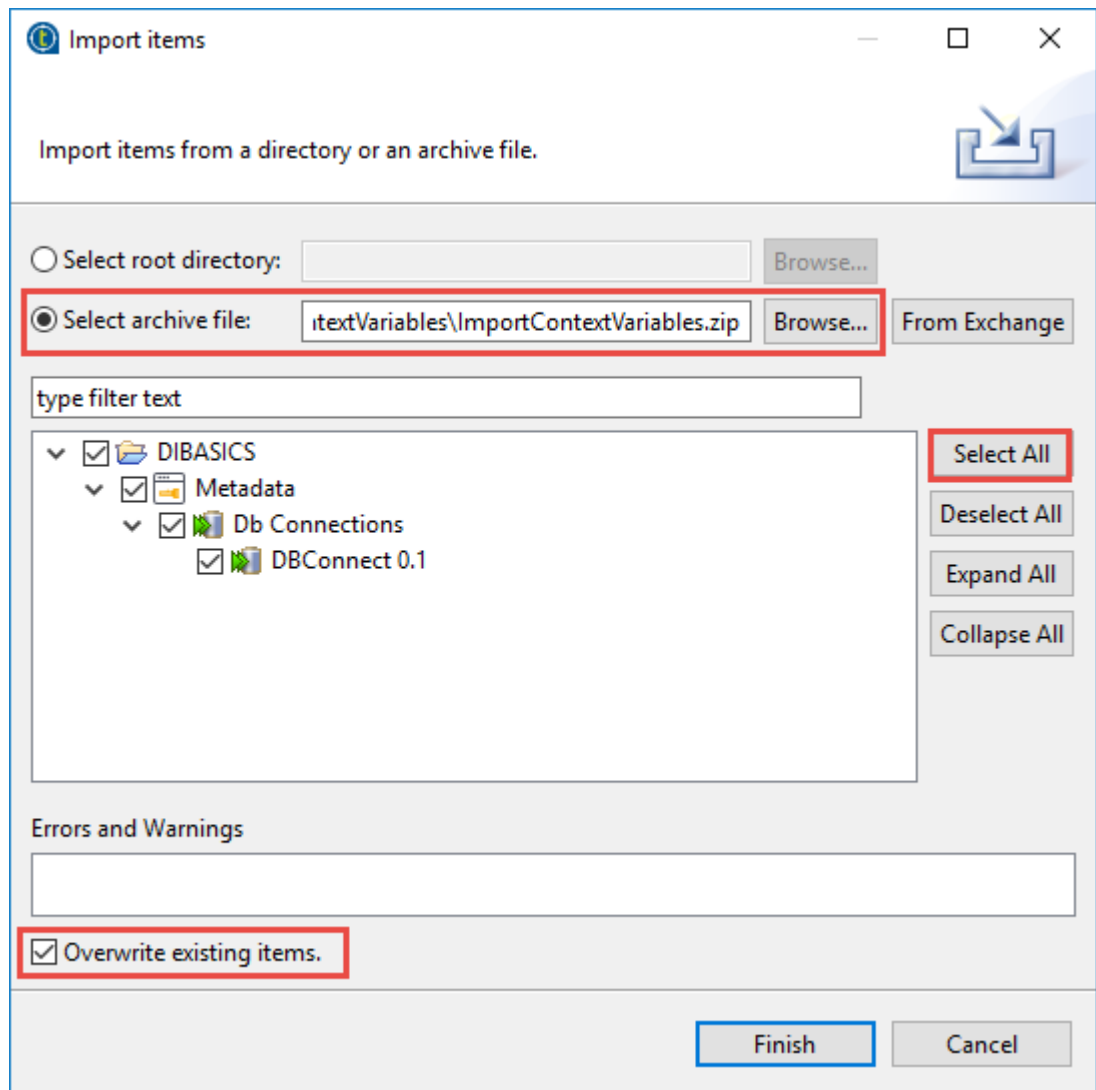


b.- En la ventana **Import item**, seleccione **Select archive file**, luego haga click en el botón **Browse**.

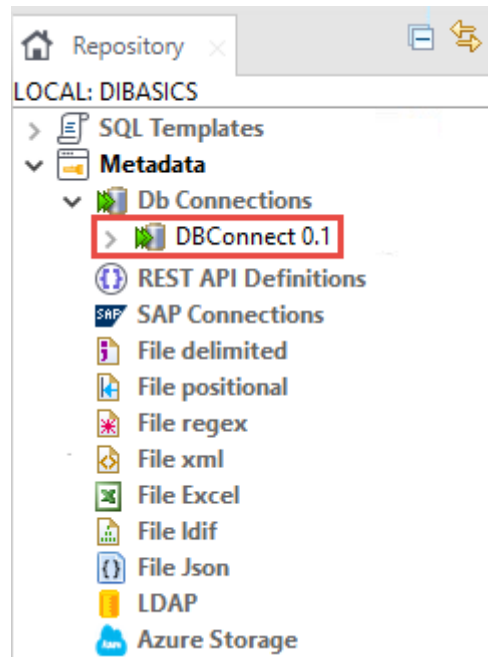
c.- Navegue hasta **C:\StudentFiles\DIBasics\ContextVariables** y abra el archivo **ImportContextVariables.zip**.

d.- Haga click en el botón **Select All**.

e.- Asegúrese de que el checkbox **Overwrite existing items** esté marcado, y presione **Finish**.



La metadata de DBConnect se ha importado al repositorio.





## Tema 2: Creación de una variable de contexto build-in

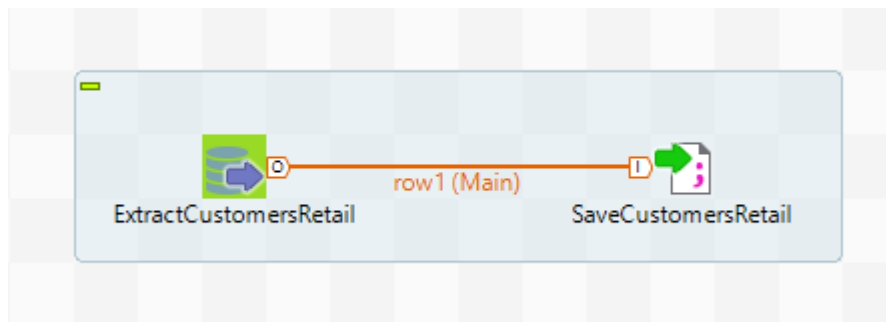
### Antes de comenzar

Este ejercicio le enseña cómo definir contextos y variables built-in para cambiar los valores de los parámetros de configuración ejecutando un Job en un contexto diferente.

Aprenderá a crear un Job para extraer datos de una base de datos en un archivo delimitado. Luego, creará una variable para obtener una ruta de archivo diferente cuando el Job se ejecute en el entorno de desarrollo o producción.

Esta variable built-in solo está disponible en el Job en el que se creó.

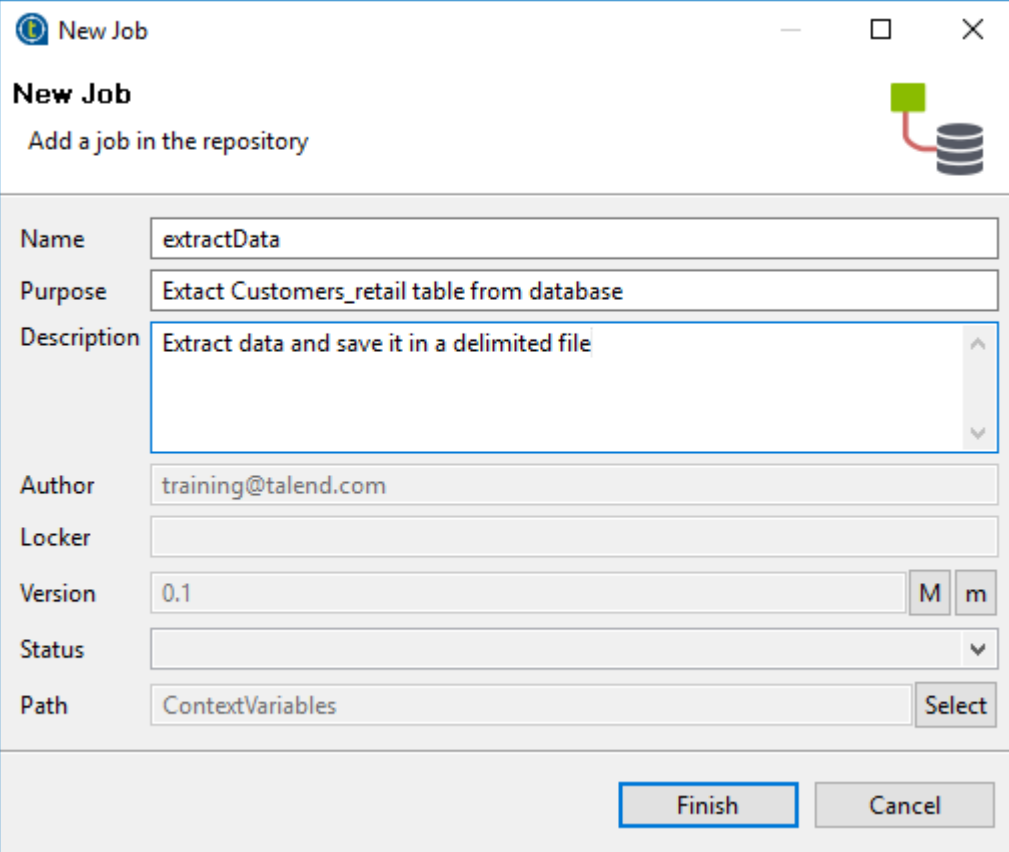
Al final, su Job debería verse así:



### Crear un Job

En un nuevo Job, usando la metadata del **Repository**, coloque un componente **tDBInput** en el workspace. luego agregue un componente **tFileOutputDelimited** para guardar localmente la información extraída de la base de datos.

**1.-** Cree un nuevo Job en **Repository > Job Designs > Standard > ContextVariables** y nómbrelo *extractData*.



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

Name: extractData

Purpose: Extract Customers\_retail table from database

Description: Extract data and save it in a delimited file

Author: training@talend.com

Locker:

Version: 0.1 M m

Status:

Path: ContextVariables Select

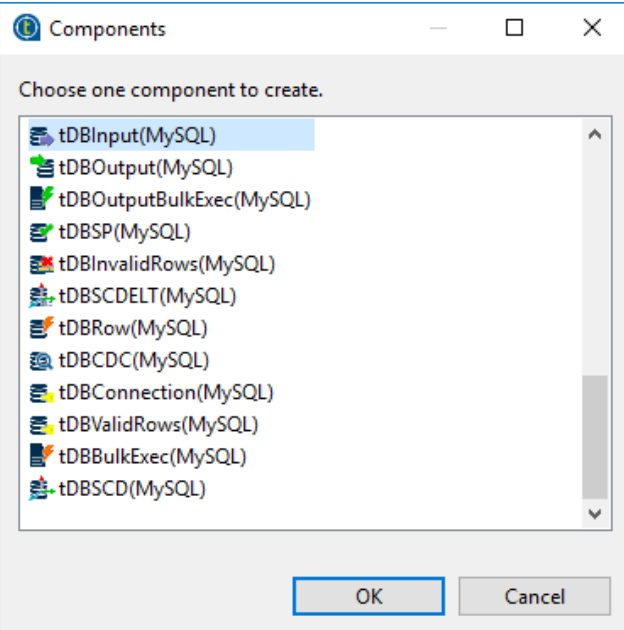
Finish Cancel

El Job se va a abrir en el Designer.

2.- Agregue un componente tDBInput.

a.- En el **Repository**, expanda la conexión **DBConnect** y arrastre el schema **customers\_retail** dentro del **Designer**.

b.- Seleccione el componente **tDBInput**.



**Components**  
Choose one component to create.

- tDBInput(MySQL)
- tDBOutput(MySQL)
- tDBOutputBulkExec(MySQL)
- tDBSP(MySQL)
- tDBInvalidRows(MySQL)
- tDBSCDELT(MySQL)
- tDBRow(MySQL)
- tDBCDC(MySQL)
- tDBConnection(MySQL)
- tDBValidRows(MySQL)
- tDBBulkExec(MySQL)
- tDBSCD(MySQL)

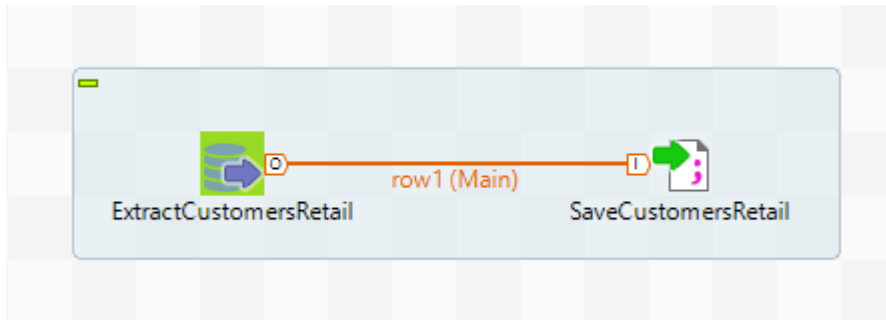
OK Cancel

c.- Renombre el componente a *ExtractCustomersRetail*.

3.- Agregue un componente tFileOutputDelimited para guardar localmente los datos extraídos.

a.- Coloque un componente **tFileOutputDelimited** en el Designer y conéctelo con **tDBInput** usando una **Main** row.

Renombre el componente a *SaveCustomersRetail*.

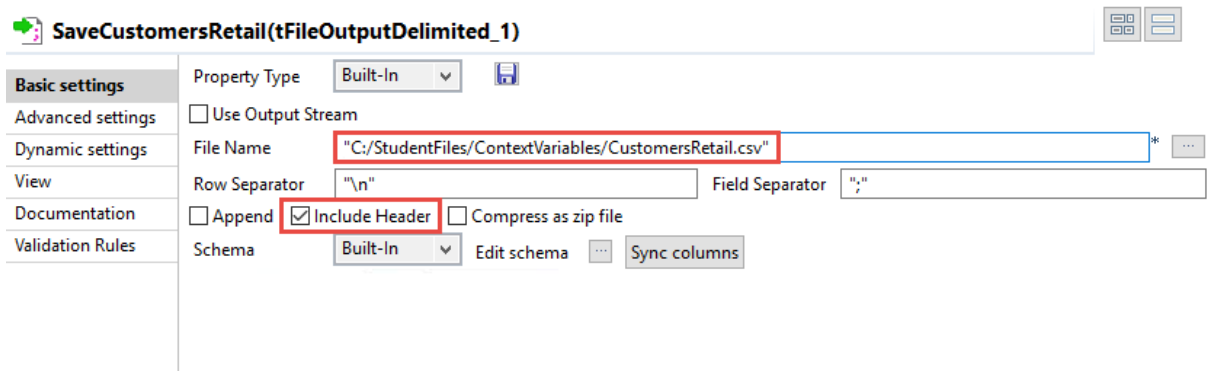


b.- Abra la pestaña **Component** y configure los parámetros:

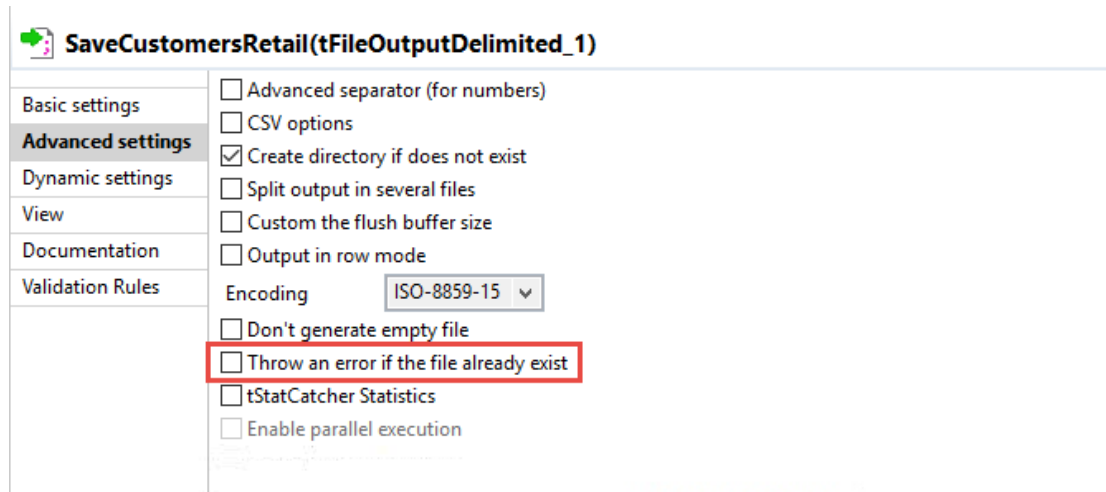
→ Para **File Name**, ingrese

"C:/StudentFiles/DIBasics/ContextVariables/CustomersRetail.csv"

→ Seleccione la checkbox **Include Header**.



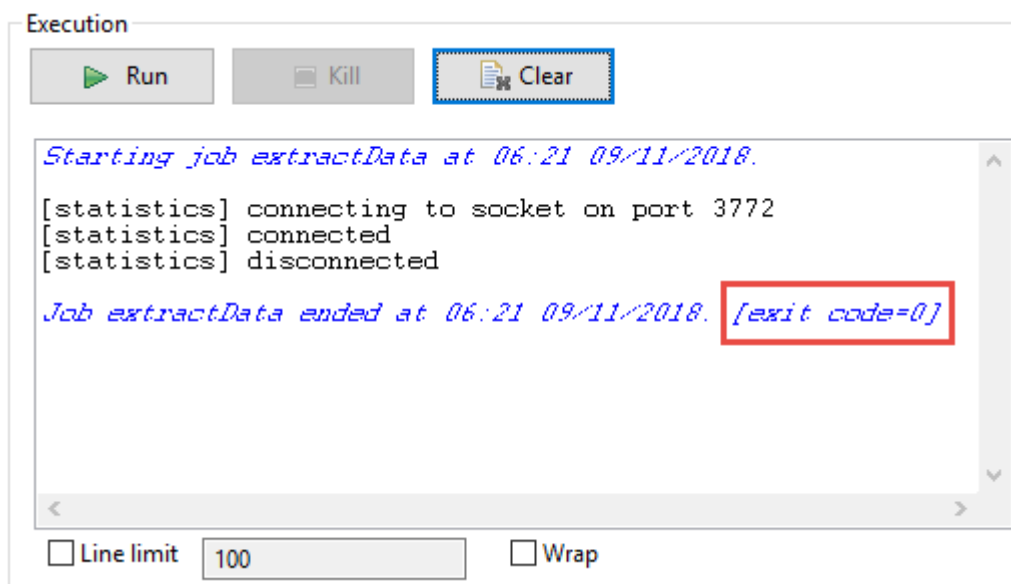
→ En la pestaña **Advanced settings**, desmarque la opción **Throw an error if the file already exist**.



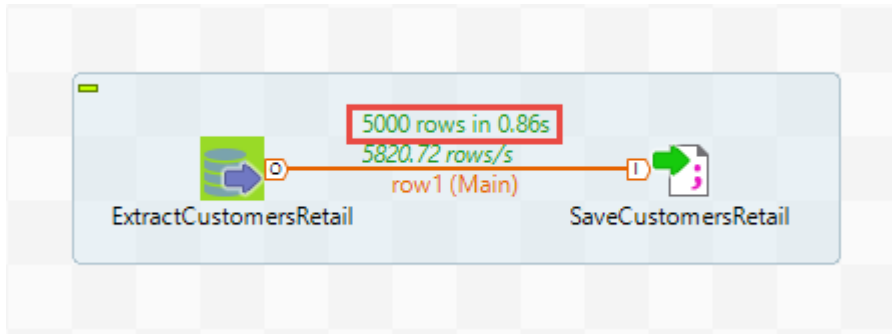
💡 En este ejercicio, ejecutará este Job varias veces en diferentes contextos. Desmarcar esta opción evita mensajes de error inútiles. Mantener esta opción habilitada es más relevante en un Job de producción.

#### 4.- Ejecute el Job.

Abra la pestaña **Run** y presione **Run**. Confirme que el Job finaliza en **[exit code=0]**.



En el **Designer**, observe que 5,000 rows fueron extraídas de la tabla.



💡 Explore el sistema de archivos para verificar que se haya creado el archivo, luego verifique su nombre y directorio.

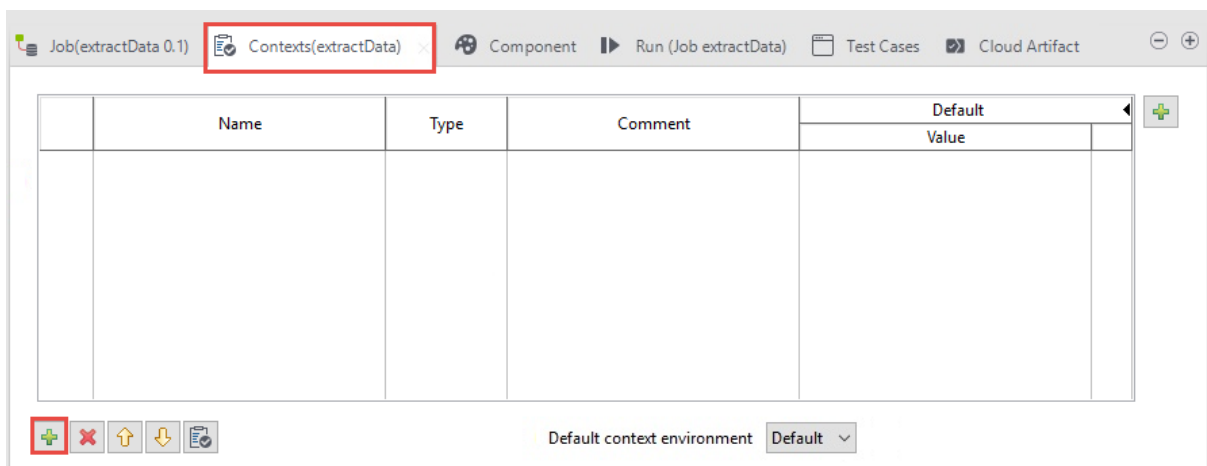
## Crear una variable

Cree una variable para definir la ruta de la salida del archivo, luego utilícela en la configuración del componente tFileOutputDelimited.

1.- Crear una variable para la ruta del archivo.

a.- Abra la pestaña **Contexts**.

b.- Para agregar una variable de contexto, en la esquina inferior izquierda, haga clic en el botón +.



c.- Enter these parameters:

- Para **Name**, ingrese *FilePath*.
- Para **Type**, seleccione **Directory**.
- En **Comment**, agregue una descripción.
- Dentro del cuadro de texto **Value**, a la derecha, haga clic en el cuadrado

para explorar el sistema de archivos y seleccione el directorio

**C:/StudentFiles/DIBasics/ContextVariables/**.

	Name	Type	Comment	Default Value
1	FilePath	Directory	Local path	C:/StudentFiles/DIBasics/ContextVariables/

La variable está lista para ser utilizada.

💡 También puede ingresar la ruta manualmente.

## 2.- Configurar la ruta de salida.

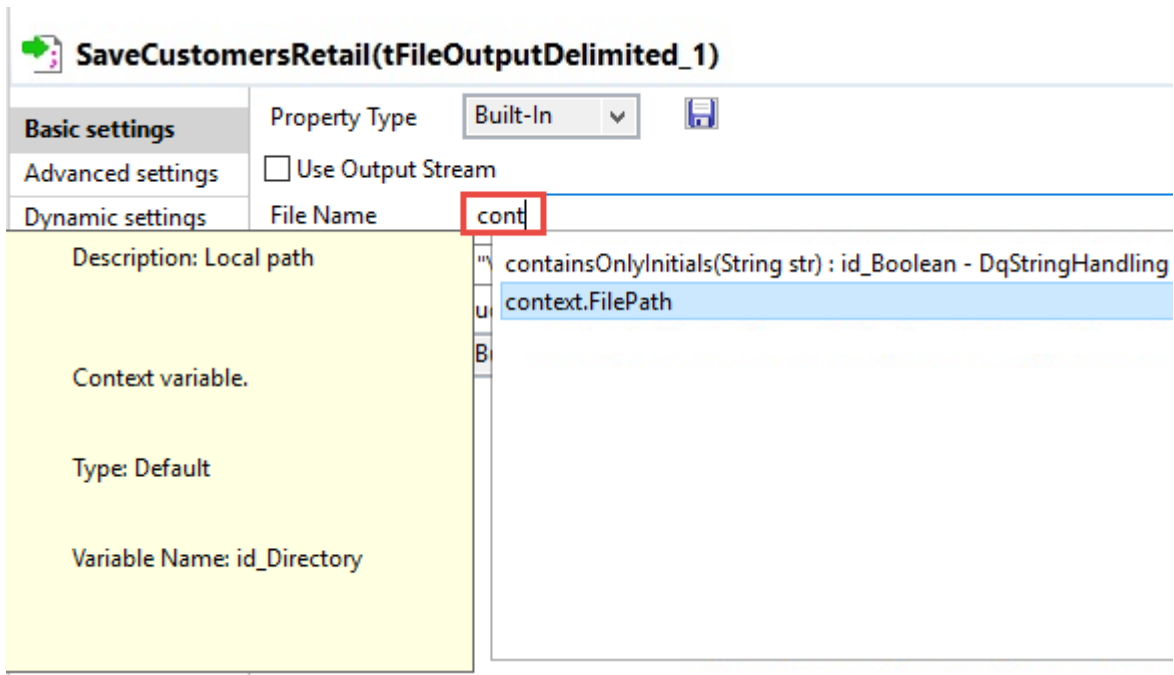
Use la variable **FilePath** para configurar la salida del componente **tFileOutputDelimited**.

a.- Haga double click en el componente **tFileOutputDelimited** para abrir la ventana **Component**.

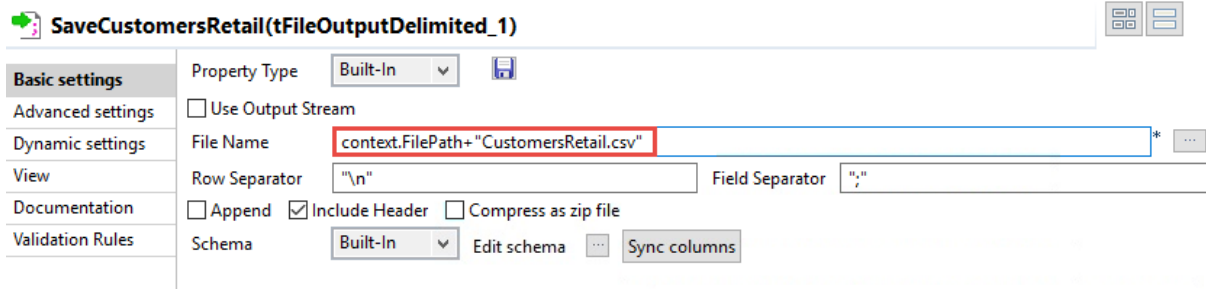
b.- Borre el texto en **File Name**.

c.- Coloque el cursor en la caja de texto **File Name** y presione **CTRL+ SPACE**.

Para mostrar las variables disponibles, comience ingresando context, luego seleccione **context.FilePath**.



d.- En el cuadro de texto **File Name**, complete el valor ingresando +, luego el nombre del archivo y la extensión:  
*context.FilePath+"CustomersRetail.csv"*



**SaveCustomersRetail(tFileOutputDelimited\_1)**

**Basic settings**

Property Type: Built-In

☐ Use Output Stream

File Name: context.FilePath+ "CustomersRetail.csv" \*

Row Separator: "\\n" Field Separator: ","

☐ Append ☒ Include Header ☐ Compress as zip file

Schema: Built-In Edit schema Sync columns

3.- Ejecute el Job.

Abra la ventana **Run** y presione **Run**. Confirme que el Job termina con **[exit code=0]**.

💡 Abra la carpeta para verificar que el archivo se haya actualizado.

## Manejo de contextos

Ha creado una variable para el contexto predeterminado. Para configurar diferentes valores para la misma variable, debe crear otros contextos.

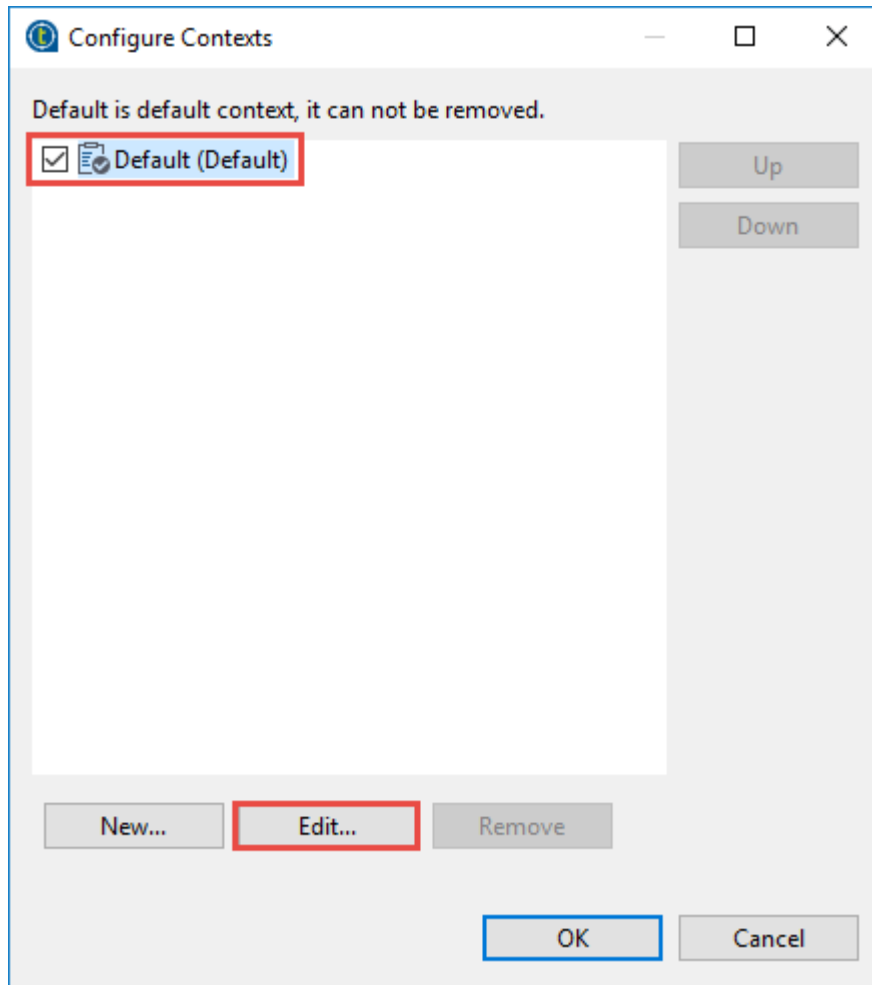
En este ejercicio, cambiará el nombre del contexto default por *Development* y creará un nuevo contexto para producción. Luego seleccionará un valor específico para la variable en producción.

1.- Configurar el contexto predeterminado y renombrarlo *Development*.

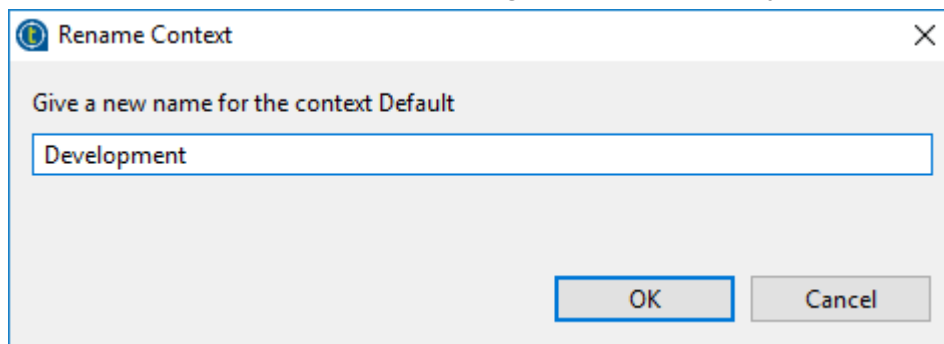
a.- En la pestaña **Contexts**, a la derecha, presione el botón **+**.

Default	
Value	
C:/StudentFiles/ContextVariables/	<input type="checkbox"/>

b.- En la ventana **Configure Contexts**, seleccione **Default** y presione **Edit**.

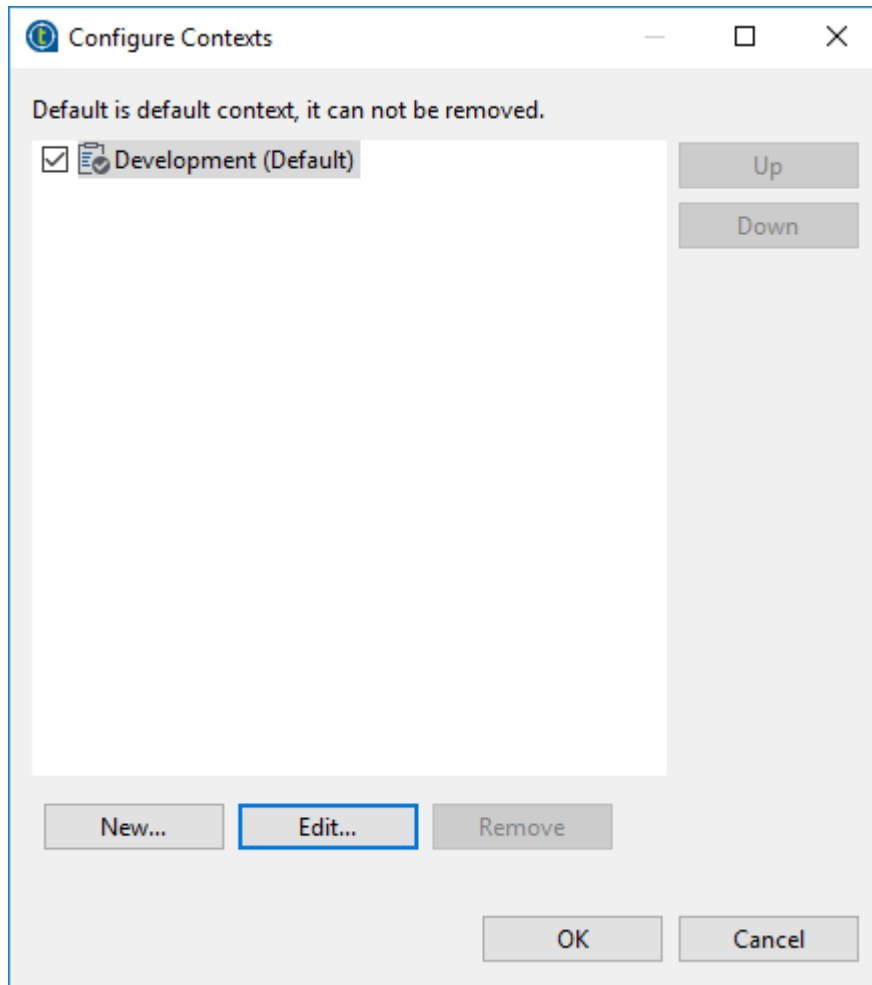


c.- En la ventana **Rename Context**, ingrese *Development* y presione **OK**.





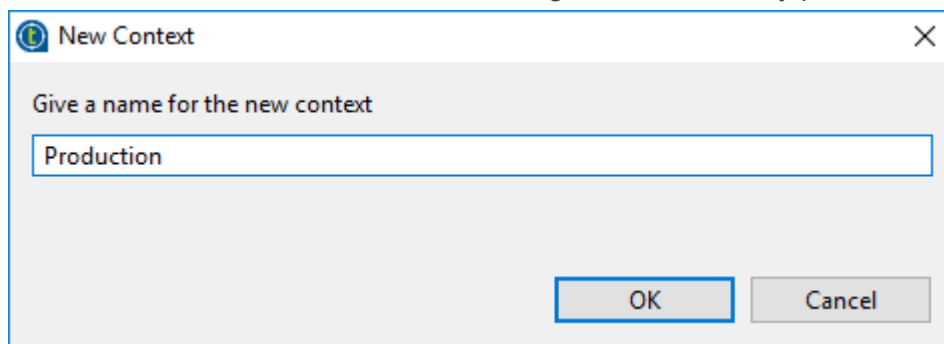
El contexto aparecerá con su nuevo nombre.



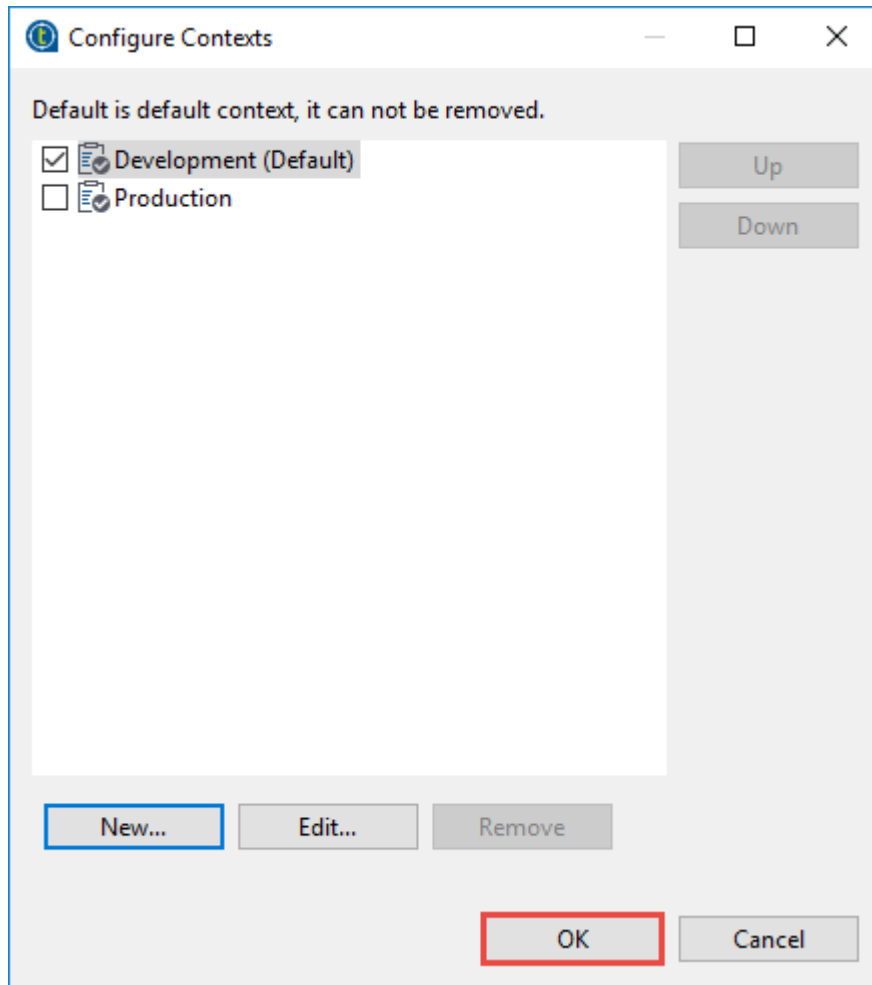
2.- En la misma ventana, crea un nuevo contexto para producción.

a.- En la ventana **Configure Contexts**, presione **New**.

b.- En la ventana **New Context**, ingrese *Production* y presione **OK**.



c.- Presione **OK**.



En la vista Contextos, el contexto Producción aparece en una nueva columna. La variable tiene los mismos valores para ambos contextos.

Development		Production	
Value		Value	
C:/StudentFiles/DIBasics/ContextVariables/	<input type="checkbox"/>	C:/StudentFiles/DIBasics/ContextVariables/	<input type="checkbox"/>

💡 Nombre sus contextos cuidadosamente. En el próximo ejercicio, creará grupos de contexto adicionales en el repositorio. La creación de contextos con exactamente el mismo nombre le permite fusionarlos en un trabajo.

### 3.- Actualice la variable.

Modifique la variable para guardar el archivo delimitado en dos subcarpetas separadas del directorio original: DEV para el contexto de desarrollo y PROD para el contexto de

producción.

a.- En la columna **Development**, actualice la ruta a *C:/StudentFiles/DIBasics/ContextVariables/DEV/*

💡 Puede escribir en el cuadro de texto o explorar el sistema de archivos y crear una nueva carpeta.

**b.-** En la columna **Production**, actualice la ruta a *C:/StudentFiles/DIBasics/ContextVariables/PROD/*

Development		Production	
Value		Value	
C:/StudentFiles/DIBasics/ContextVariables/DEV/	<input type="checkbox"/>	C:/StudentFiles/DIBasics/ContextVariables/PROD/	<input type="checkbox"/>

4.- Ejecuta el Job usando los dos contextos que creó.

**a.-** Abra la pestaña **Run**. En la lista desplegable a la derecha de la consola, el contexto **Development** está seleccionado de forma predeterminada.

Presione **Run**. Confirme que el Job finaliza con **[exit code=0]**.

### Job extractData

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

Execution

Run
 Kill
 Clear

```
Starting job extractData at 13:44 10/11/2018.
```

```
[statistics] connecting to socket on port 3970
```

```
[statistics] connected
```

```
[statistics] disconnected
```

```
Job extractData ended at 13:44 10/11/2018. [exit code=0]
```

☐ Line limit 
☐ Wrap

**Development**

Name
FilePath

>>

**b.-** En la lista desplegable, seleccione el contexto **Production**.

Presione **Run**. Confirme que el Job finaliza con **[exit code=0]**.

**Job extractData**

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

Execution

```

Starting job extractData at 13:46 10/11/2018.
[statistics] connecting to socket on port 3665
[statistics] connected
[statistics] disconnected

Job extractData ended at 13:46 10/11/2018. [exit code=0]
                
```

☐ Line limit 
☐ Wrap

Production

Name	FilePath

c.- Examine el sistema de archivos y confirme que se han creado las dos subcarpetas y que ambas almacenan una versión del archivo.

← → ▾ ↑

This PC > Local Disk (C:) > StudentFiles > DIBasics > ContextVariables > DEV

▾ DIBasics
 

▾ ContextVariables
 

DEV

PROD

ImportContextVariables.zip

Name

CustomersRetail.csv

## Tema 3: Conexión a bases de datos usando variables de contexto

### Antes de comenzar

---

Este ejercicio explica cómo usar variables de contexto cuando se trabaja con bases de datos.

El primer ejercicio le muestra cómo crear un nuevo grupo de contexto para almacenar la configuración de conexión de la base de datos como variables. Este grupo de contexto se crea en el repositorio y puede ser reutilizado por cualquier trabajo en el proyecto.

Luego, actualiza las variables para conectarse a la base de datos sandbox cuando usa el contexto de development y a la base de datos training\_data cuando usa el contexto de production.

Por último, ejecuta el Job utilizando los contextos de desarrollo y producción.

### Creación de un contexto de repositorio para la conexión de base de datos

---

En este ejercicio, editará la metadata de conexión de base de datos existente en el repositorio para exportar las credenciales de conexión en un nuevo contexto de repositorio.

💡 Las bases de datos de desarrollo y producción se almacenan juntas en su entorno de entrenamiento. Ambos contienen la tabla Customers\_Retail con exactamente los mismos datos.

1.- Exportar la metadata de la base de datos como contexto.

a.- En **Repository** expanda **Metadata > Db connections**, haga click derecho en **DBConnect** y seleccione **Edit connection**. Haga click en **Next** en la ventana **Database**

## Connection.

Database Connection

Update Database Connection - Step 1/2

Update properties

Name

DBConnect

Purpose

metadata to connect to database

Description

MySQL database

Author

training@talend.com

Locker

training@talend.com

Version

0.1

Mm

Status

Path

< Back

Next >

Finish

Cancel


**b.-** Estos son los ajustes de conexión para la base de datos de desarrollo. Los usaste anteriormente en tu Job.

Para exportar esta configuración como una colección de variables de contexto, presione

## Export as context.

Database Connection

### Update Database Connection - Step 2/2

 You must press the Check Button to check the Database Setting



DB Type

MySQL

Db Version

MySQL 5

String of Connection

jdbc:mysql://localhost:3306/training?noDatetimeStringSync=true

Login

talend

Password

•••••

Server

localhost

Port

3306


DataBase

training

Additional parameters

noDatetimeStringSync=true

Test connection



Export as context

Revert Context

< Back

Next >

Finish

Cancel

c.- En la ventana **Create / Reuse a context group**, seleccione **Create a new repository context** y presione **Next**.

d.- Para **Name**, ingrese *DBConnectContext*, y agregue un **Purpose** y una **Description**, luego presione next **Next**.

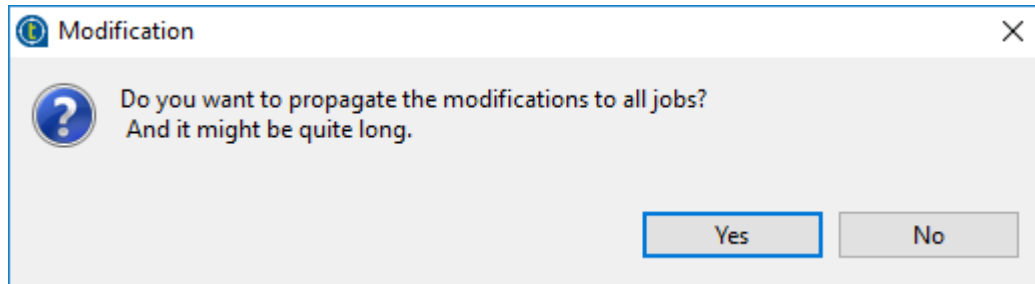
Se han creado seis variables de contexto para el contexto predeterminado.

	Name	Type	Comment	Default Value	
1					
2	DBConnect_Login	String		talend	<input type="checkbox"/>
3	DBConnect_Port	String		3306	<input type="checkbox"/>
4	DBConnect_Password	Password		*****	<input type="checkbox"/>
5	DBConnect_AdditionalProperties	String		noDatetimeStringSync=true	<input type="checkbox"/>
6	DBConnect_Server	String		localhost	<input type="checkbox"/>
7	DBConnect_Database	String		training	<input type="checkbox"/>

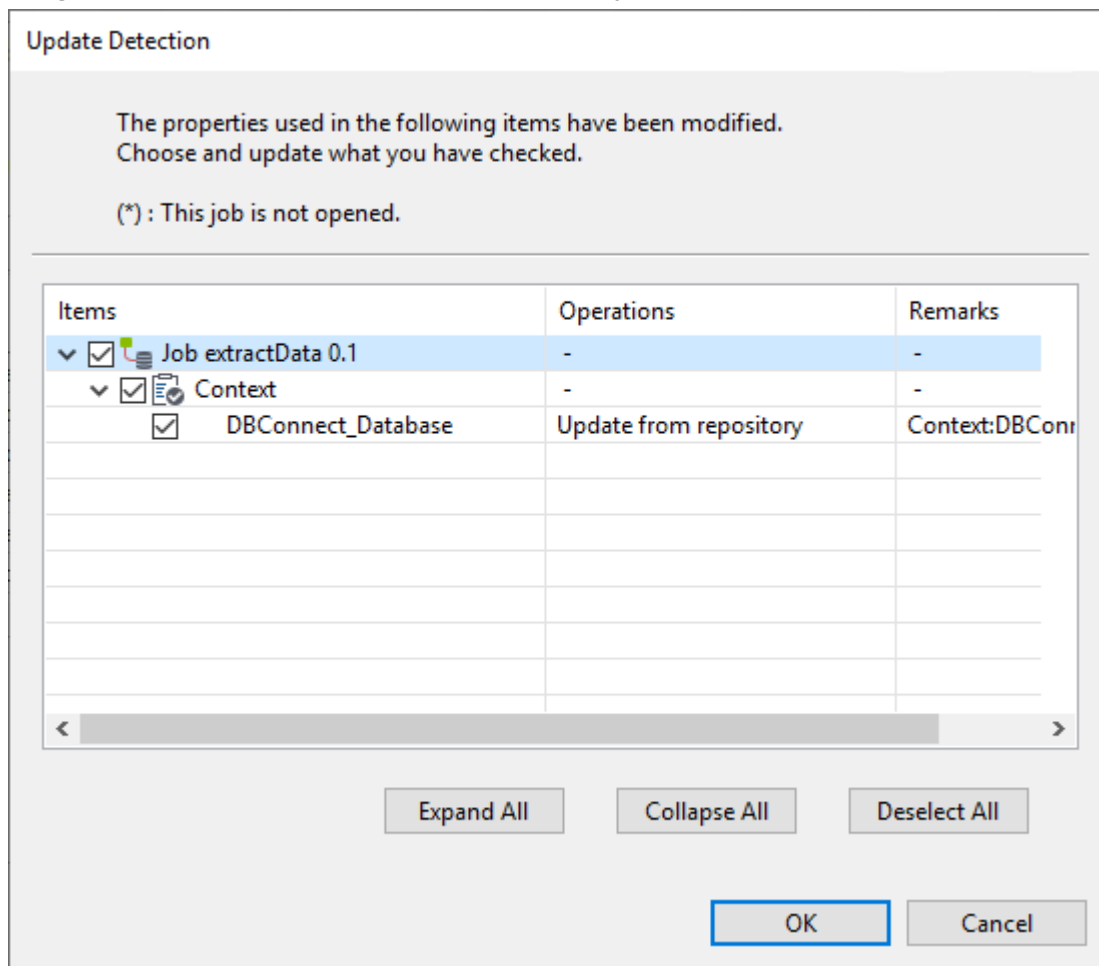


⚠ Para mantener la coherencia con los contextos utilizados en su Job, debe cambiar ,dentro del contexto de repositorio, el nombre del contexto *Default* por *Development* y crear un segundo contexto llamado *Production*. ⚠

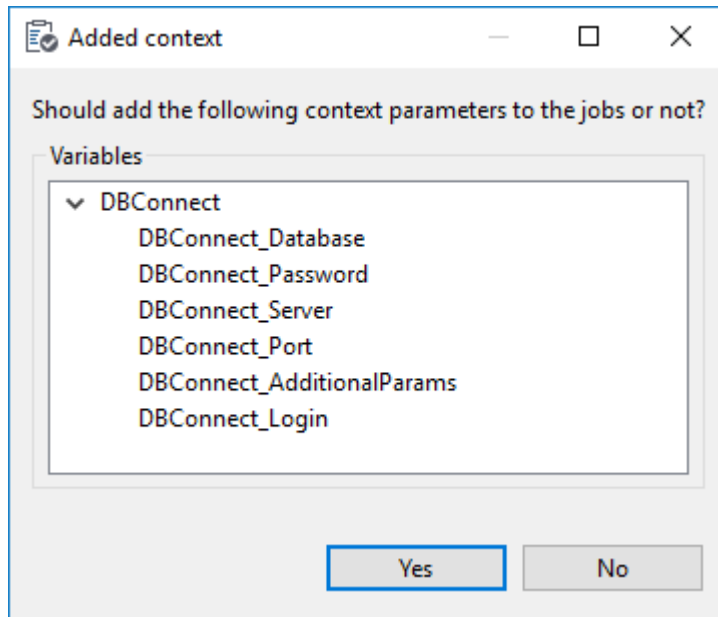
3.- Al terminar de modificar el contexto de desarrollo verá una ventana que le preguntará si quiere propagar los cambios en su Job, presione **Yes**.



Asegúrese de seleccionar todas las instancias y presione **OK**.




Cuando se le pregunte si desea agregar los parámetros de contexto a los Jobs, presione **Yes**.




Su Job hereda directamente las nuevas variables de contexto.

#### 4.- Finalizar la creación del grupo de contexto.

Volviendo a la ventana **Database Connection** , observe que los valores han sido reemplazados con variables de contexto. Haga click en **Cancel**.


**Database Connection**
—
□
×

**Update Database Connection - Step 2/2**


DB Type

MySQL

Db Version

MySQL 5

String of Connection

jdbc:mysql://localhost:3306/sandbox?noDatetimeStringSync=true

Login

context.DBConnect\_Login

Password

context.DBConnect\_Password

Server

context.DBConnect\_Server

Port

context.DBConnect\_Port

DataBase

context.DBConnect\_Database

Additional parameters

context.DBConnect\_AdditionalParams

Test connection

v

Export as context

Revert Context

[How to install a driver](#)

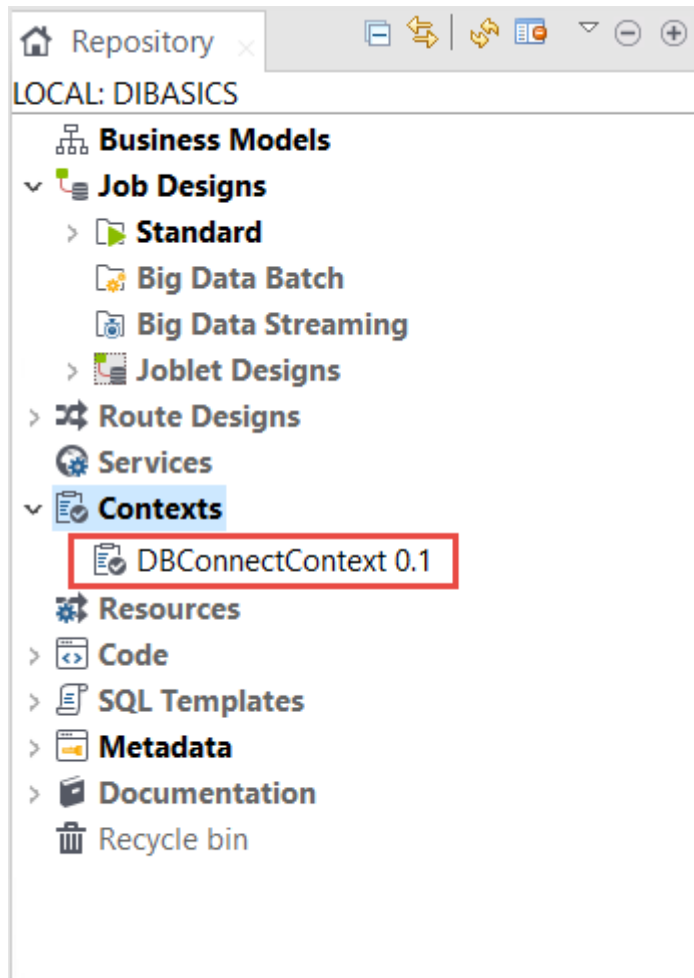
< Back

Next >

Finish

Cancel

El nuevo context group es visible en el repositorio.



## Actualizar el context group


En este ejercicio, actualizará el grupo de contexto en el repositorio para poder conectar su Job a la base de datos de producción o desarrollo, según el contexto seleccionado.

1.- Editar el context group.

Actualice la variable **DBConnect\_Database** para conectarse a la base de datos de datos training\_data cuando utilice el contexto de producción.

a.- En el Repositorio, expanda **Contexts**, haga clic con el botón derecho en DBConnectContext y seleccione **Edit context group**. Luego presione **Next** en la ventana

## Create / Edit a context group.

 Create / Edit a context group

Step 1 of 2

Add any required information

Name

DBConnectContext

Purpose

Database variables

Description

connection to development and production database

Author

user@talend.com

Locker

user@talend.com

Version

0.1

M

m

Status

Path

< Back

Next >

Finish

Cancel

b.- Edite el valor de la variable **DBConnect\_Database** para el contexto **Production**, bórralo, e ingrese *production*.

Create / Edit a context group

**Step 2 of 2**  
Define the contexts, variables and values

	Name	Type	Comment	Development		Production	
				Value	<input type="checkbox"/>	Value	<input type="checkbox"/>
1	_Database	String		training	<input type="checkbox"/>	production	<input type="checkbox"/>
2	_Login	String		talend	<input type="checkbox"/>	talend	<input type="checkbox"/>
3	_AdditionalP	String		noDatetimeStringSync=tru	<input type="checkbox"/>	noDatetimeStringSync=tru	<input type="checkbox"/>
4	_Password	Password		*****	<input type="checkbox"/>	*****	<input type="checkbox"/>
5	_Server	String		localhost	<input type="checkbox"/>	localhost	<input type="checkbox"/>
6	_Port	String		3306	<input type="checkbox"/>	3306	<input type="checkbox"/>

Default context environment: Development

< Back   Next >   **Finish**   Cancel

c.- Click en **Finish**.

Para propagar los cambios en el Job, haga click en **Yes**.

**Modification**

Do you want to propagate the modifications to all jobs?  
And it might be quite long.

**Yes**   **No**

Asegúrese de haber seleccionado todos los ítems y presione **OK**.

Update Detection

The properties used in the following items have been modified.  
Choose and update what you have checked.

(\*) : This job is not opened.

Items	Operations	Remarks
<input checked="" type="checkbox"/> Job extractData 0.1	-	-
<input checked="" type="checkbox"/> Context	-	-
<input checked="" type="checkbox"/> DBConnect_Database	Update from repository	Context:DBCon

Expand All

Collapse All

Deselect All

OK

Cancel

2.- Confirme que se actualizaron las variables en la pestaña **Contexts** y en la configuración de **tDBInput**.

a.- El Job extractData debería estar abierto. Si no lo está, ábralo.

	Name	Type	Comment	Development		Production	
				Value		Value	
1	FilePath	Directory	File path	C:/StudentFiles/DIBasics/ContextVariables/DE	<input type="checkbox"/>	C:/StudentFiles/DIBasics/ContextVariables/PROI	<input type="checkbox"/>
2	DBConnectContext (from repository c						
3	DBConnect_Database	String		training	<input type="checkbox"/>	production	<input type="checkbox"/>
4	DBConnect_Password	Password		*****	<input type="checkbox"/>	*****	<input type="checkbox"/>
5	DBConnect_Server	String		localhost	<input type="checkbox"/>	localhost	<input type="checkbox"/>
6	DBConnect_Port	String		3306	<input type="checkbox"/>	3306	<input type="checkbox"/>
7	DBConnect_AdditionalParams	String		noDatetimeStringSync=true	<input type="checkbox"/>	noDatetimeStringSync=true	<input type="checkbox"/>
8	DBConnect_Login	String		talend	<input type="checkbox"/>	talend	<input type="checkbox"/>

Default context environment Develop

Las nuevas variables DBConnect se agregaron a la variable FilePath que creó anteriormente.

💡 Los contextos con exactamente el mismo nombre se fusionan en el Job.

**b.- En Designer, haga doble click en tDBInput para abrir la ventana Component.**

**ExtractCustomersRetail(tDBInput\_1)(MySQL)**

<b>Basic settings</b>	Database	MySQL	Apply
Advanced settings	Property Type	Repository	DB (MYSQL):DBConnect
Dynamic settings	DB Version	MySQL 5	
View	<input type="checkbox"/> Use an existing connection		
Documentation	Host	context.DBConnect_Server	Port context.DBConnect_Port Database context.DBConnect_Database
Validation Rules	Username	context.DBConnect_Login	Password context.DBConnect_Password
	Schema	Repository	DB (MYSQL):DBConnect - customers_retail Edit schema
	Table Name	"customers_retail"	
	Query Type	Built-In	Guess Query Guess schema
	Query	<pre>"SELECT `customers_retail`.`id`, `customers_retail`.`CustomerName`, `customers_retail`.`CustomerAddress`, `customers_retail`.`idState`,</pre>	

Todos los ajustes de conexión se definen mediante variables.

**3.- Ejecute el Job nuevamente usando los dos contextos.**

**a.- Abra la ventana Run. Seleccione el contexto Development.**  
Presione **Run**. El Job debe finalizar en **[exit code=0]**.

**b.- Seleccione el contexto Production.**  
Presione **Run**. El Job debe finalizar en **[exit code=0]**.

**c.- Examine la carpeta y confirme que los dos archivos estén actualizados.**



## Tema 4: Creación de un Context Group en el repositorio

### Antes de comenzar

En los ejercicios anteriores, creó una variable de contexto built-in y un grupo de contexto en el repositorio para almacenar la configuración de conexión de la base de datos.

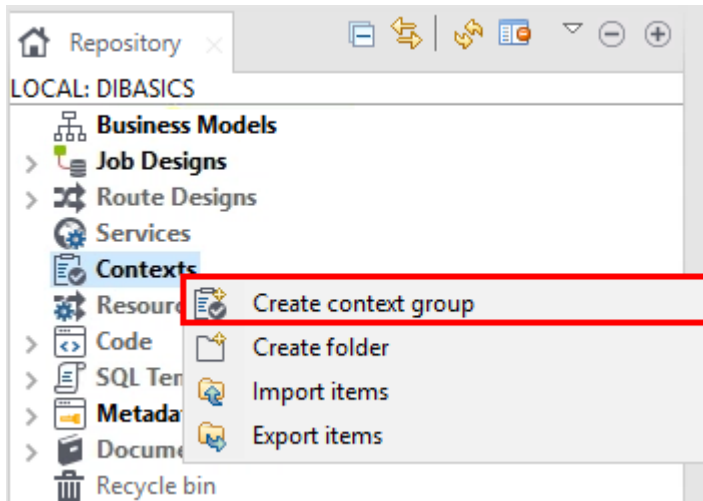
En este ejercicio, creará un nuevo grupo de contexto en el repositorio.

En su trabajo, utiliza esta variable para limitar el número de filas seleccionadas en una consulta. Solo en el contexto de desarrollo, establece una opción para solicitar el valor de la variable.

### Crear un context group en el repositorio

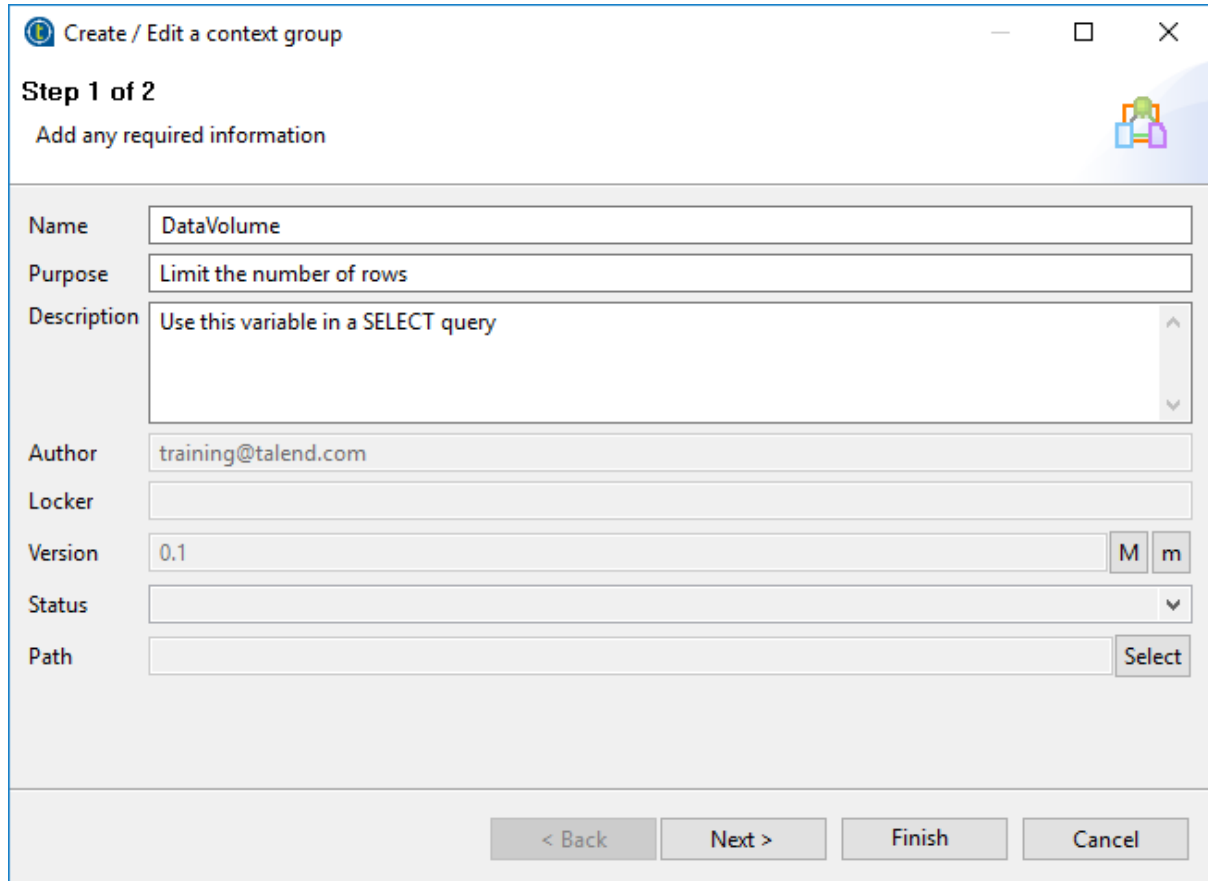
1.- Para organizar sus variables de contexto, cree un nuevo grupo de contexto en el Repositorio.

a.-En el Repositorio, haga clic con el botón derecho en **Contexts** y seleccione **Create context group**.



b.- Para **Name**, ingrese *DataVolume*. Ingrese **Purpose** y **Description** y presione

Next.



Create / Edit a context group

Step 1 of 2

Add any required information

Name: DataVolume

Purpose: Limit the number of rows

Description: Use this variable in a SELECT query

Author: training@talend.com

Locker:

Version: 0.1

Status:

Path:

M m

Select

< Back Next > Finish Cancel

2.- Agregar una nueva variable de contexto.

Defina una nueva variable de contexto para usar la función LIMIT en una consulta SELECT

a.- En la esquina inferior izquierda de la tabla, haga clic en + para agregar una variable.

b.- Configure la variable usando estos parámetros:

- Para **Name**, ingrese *Limit*.
- Para **Type**, seleccione **String**.
- Especifique un **Comment** que describa la variable.
- Para **Value**, ingrese *LIMIT 1000*.

Create / Edit a context group

**Step 2 of 2**  
Define the contexts, variables and values

	Name	Type	Comment	Default
1	Limit	String	Number of selected rows	LIMIT 1000

Default context environment: Default

< Back Next > Finish Cancel

### 3.- Administrar los contextos.

Vuelva a cambiar el nombre del contexto **Default** por *Development* y cree un nuevo contexto **Production**.

Development	Production
Value	Value
LIMIT 1000	LIMIT 1000

El contexto Producción aparece en una nueva columna. La variable tiene el mismo valor para ambos contextos.

### 4.- Actualizar la variable.

Elimine la limitación al consultar la base de datos de producción y agregue un aviso al consultar la base de datos de desarrollo.

a.- Para el contexto de Producción, elimine el valor de la variable y déjelo vacío.

b.- Para el contexto de Desarrollo, a la derecha del valor LIMIT 1000, seleccione la casilla de verificación.

c.- Actualice el **Prompt** ingresando *Enter a limit*.

Create / Edit a context group

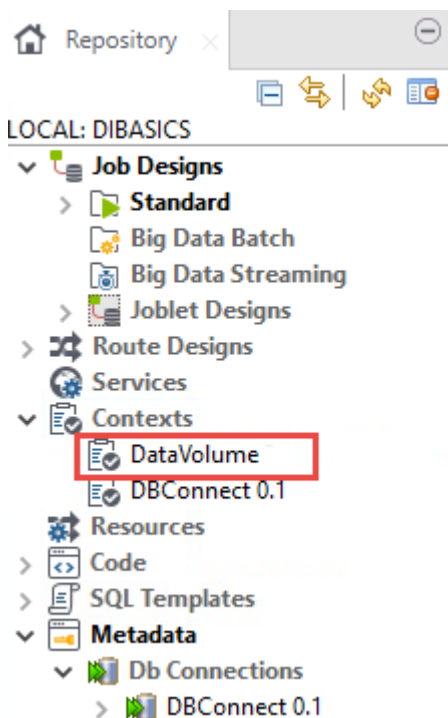
**Step 2 of 2**  
Define the contexts, variables and values

	Comment	Development		Production	
		Value	Prompt	Value	
1	Number of selected rows	LIMIT 1000	<input checked="" type="checkbox"/>	Enter a limit	<input type="checkbox"/>

Default context environment: Development

< Back   Next >   Finish   Cancel

d.- Presione **Finish**.



El nuevo grupo de contexto aparece en el repositorio.

💡 Nunca active el indicador en el contexto de producción, ya que el Job debe ejecutarse sin intervención humana.

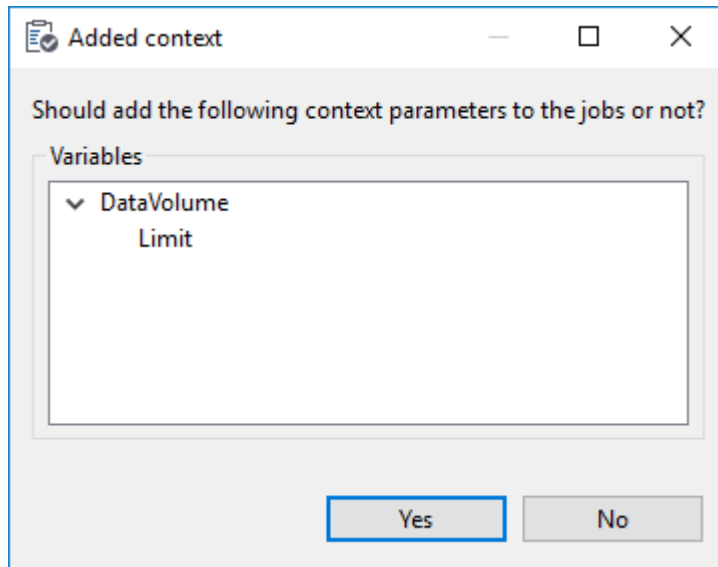
## Agregar variables de contexto a un Job

En este ejercicio, agregará el grupo de contexto a un Job, usará la variable en la consulta tDBInput y ejecutará el Job en los contextos de desarrollo y producción..

1.- En el Job abierto, seleccione el grupo de contexto y colóquelo en el **Designer**.

a.- En **Repository**, click **Contexts > DataVolume** y colóquelo en el **Designer**.

b.- En el cuadro de diálogo **Added context**, click **Yes**.



💡 Puede lograr el mismo resultado haciendo clic en el botón **Contexts** debajo de la tabla **Variables** en la pestaña **Contexts**.

La nueva variable aparece en la pestaña Contexts. Los contextos de Desarrollo y Producción para el grupo DataVolume se fusionaron con los otros contextos con los mismos nombres.

	Name	Type	Comment	Development		Production	
				Value	Prompt	Value	
1	FilePath	Directory	Local path	C:/StudentFiles/DIBasics/ContextVariab	FilePath?	C:/StudentFiles/DIBasics/ContextVariab	
2	DataVolume (from repository context)						
3	Limit	String		LIMIT 1000	Enter a limit		
4	DBConnectContext (from repository)						
5	DBConnect_AdditionalParams	String		noDatetimeStringSync=true	DBConnect_AdditionalParams?	noDatetimeStringSync=true	
6	DBConnect_Database	String		sandbox	DBConnect_Database?	training_data	
7	DBConnect_Login	String		talend	DBConnect_Login?	talend	
8	DBConnect_Password	Password		*****	DBConnect_Password?	*****	
9	DBConnect_Port	String		3306	DBConnect_Port?	3306	
10	DBConnect_Server	String		localhost	DBConnect_Server?	localhost	

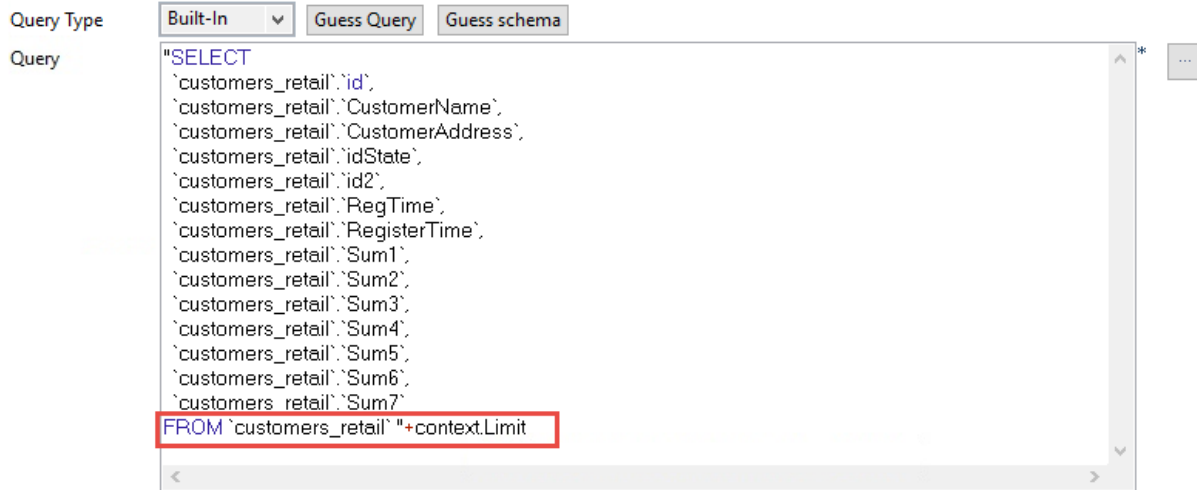


Default context environment Development

2.- Agregue la variable a la consulta tDBInput.

a.- En **Designer**, haga doble click en **tDBInput**.

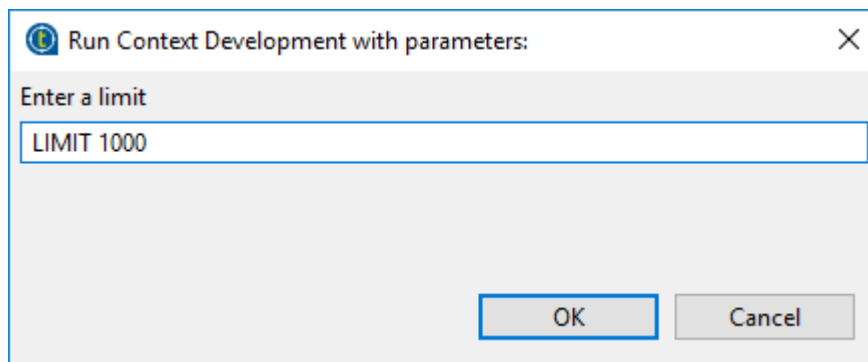
b.- En la **Query**, agregue un espacio, y luego ingrese *+context.Limit*.



3.- Ejecute el Job usando los dos contextos.

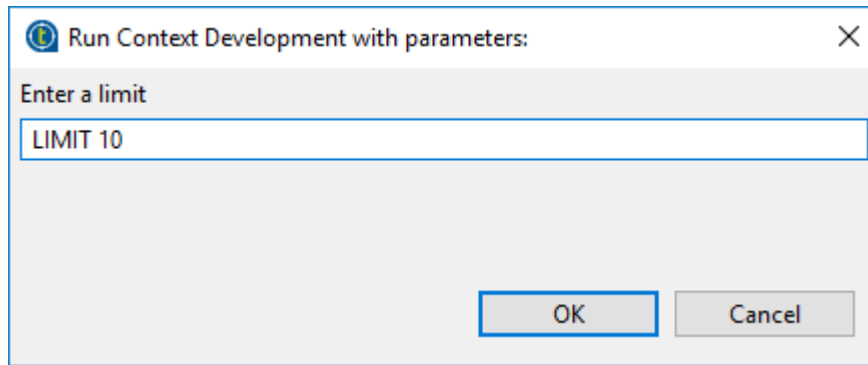
a.- Abra la pestaña **Run**. Seleccione el contexto **Development**

b.- presione **Run**.



El mensaje prompt aparecerá en pantalla.

c.- Actualice el límite en el cuadro de texto reemplazando 1000 con 10.



Confirme que la ejecución finalizó en **[exit code=0]**.

d.- Seleccione el contexto **Production**

Presione **Run**. Confirme que la ejecución finalizó en **[exit code=0]**.

e.- Examine la carpeta y confirme que los dos archivos estén actualizados.

El archivo en el directorio DEV contiene solo 10 filas de datos.

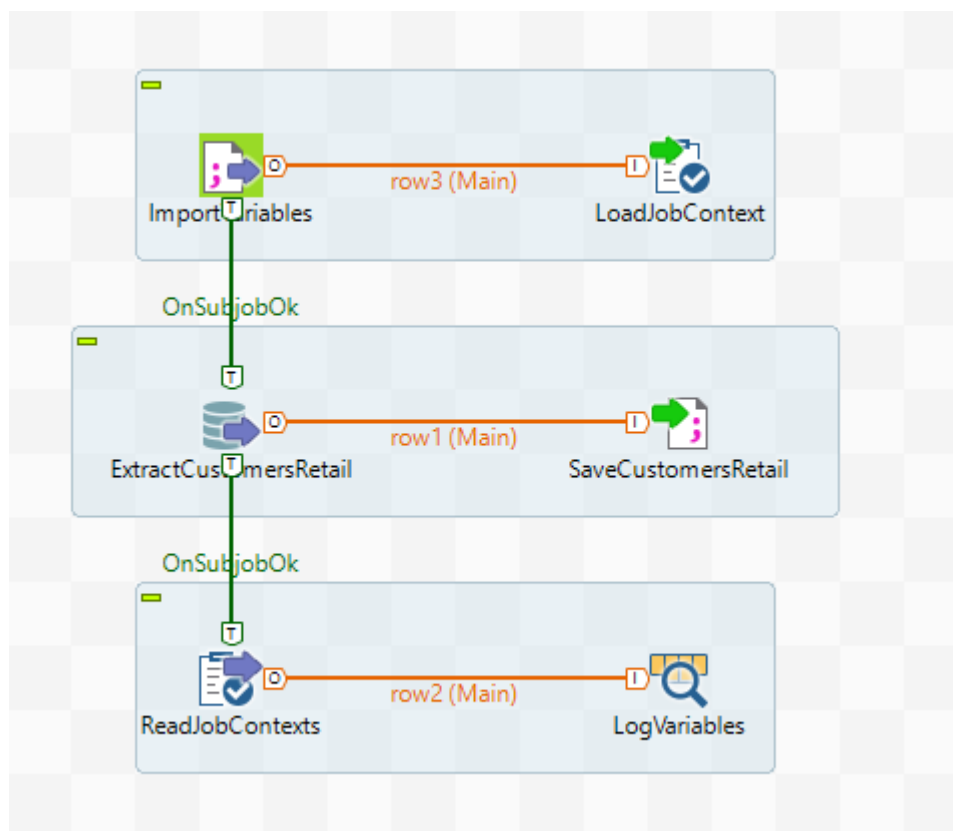
## Tema 5: Carga de variables de contexto desde un flujo (flow)

### Antes de comenzar

Este ejercicio le enseña cómo actualizar dinámicamente las variables en un Job cargando sus valores desde un flujo de datos.

En este ejercicio exportará las variables de un Job a un archivo delimitado. Luego actualiza algunos valores en este archivo y lo reutiliza para sobrescribir valores de variables en el contexto de producción.

El Job debe verse así:





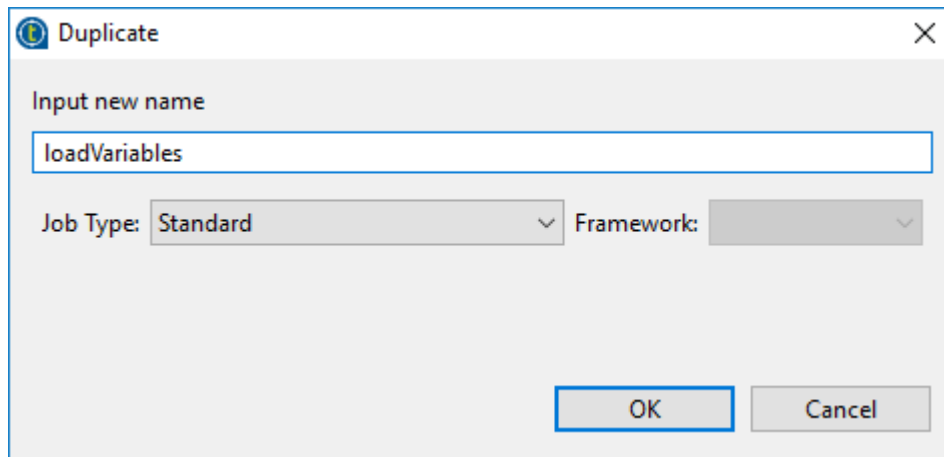
## Exportación de variables a un archivo

En este ejercicio, exportará las variables para el Job exportData Job a un archivo delimitado. Este archivo se reutiliza más tarde para actualizar dinámicamente las variables.

1.- Para mantener el Job **exportData** sin modificaciones, duplíquelo y ábralo en el **Designer**.

a.- En el **Repository**, expanda **Job Designs > Standard > ContextVariables**, click derecho en **extractData**, y seleccione **Duplicate**.

b.- En la ventana **Duplicate**, para **Input new name**, ingrese *loadVariables* y presione **OK**.

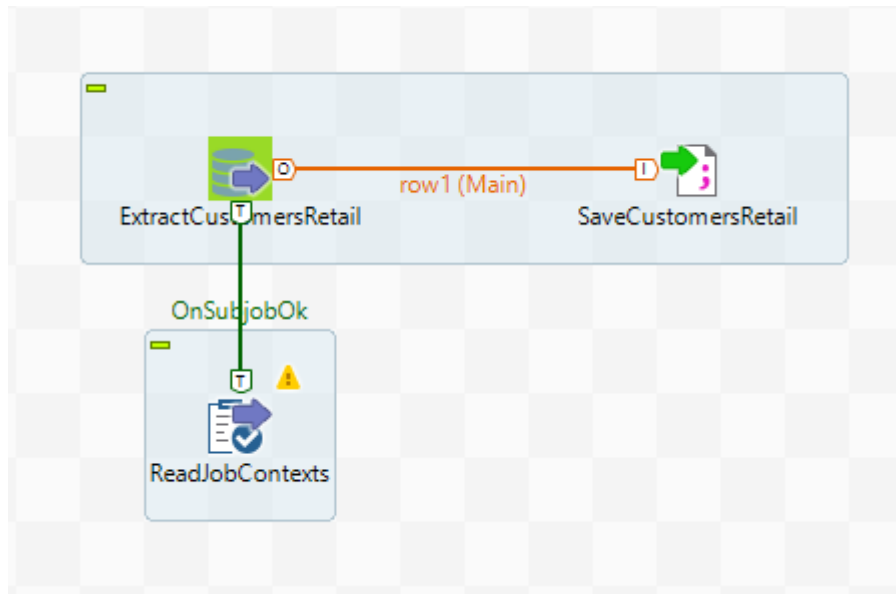


c.- Abra el Job **loadVariables**.

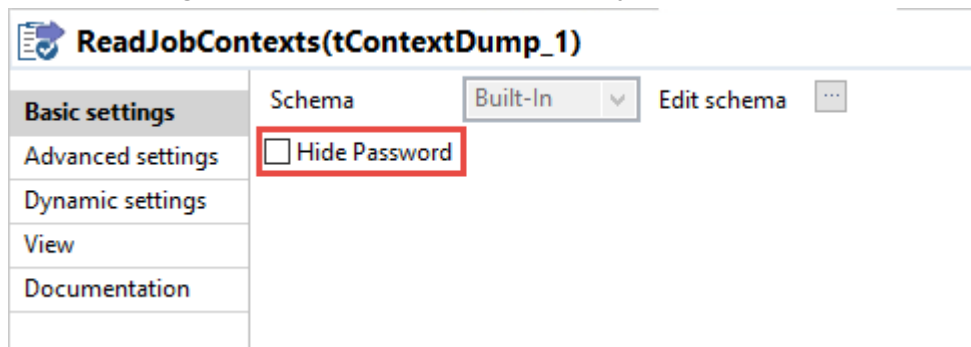
2.- Agregue un componente **tContextDump** para leer las variables del Job.

a.- En **Designer**, coloque un componente **tContextDump** debajo de **tDBInput** y renómbrelo *ReadJobContext*.

b.- Conecte **tDBInput** a **tContextDump** usando trigger **On Subjob OK**.



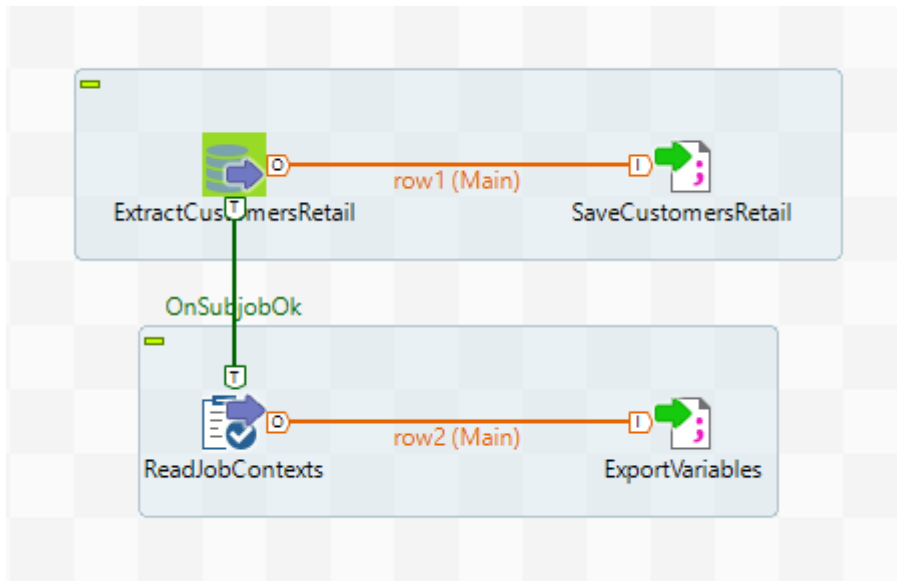
c.- Haga doble click en **tContextDump** y desmarque el check box **Hide Password**.



3.- Agregue un componente **tFileOutputDelimited** para exportar las variables del Job al archivo local.

a.- A la derecha de **tContextDump**, coloque un componente **tFileOutputDelimited** y nómbrelo *ExportVariables*.

b.- Conecte **tContextDump** a **tFileOutputDelimited** usando **Main row**.

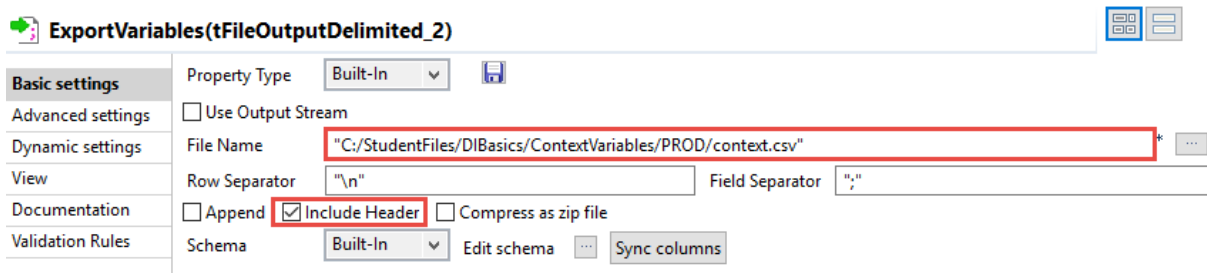


c.- Haga doble click en **tFileOutputDelimited** e ingrese los siguientes parámetros:

→ Para **File Name**, ingrese

*C:/StudentFiles/DIBasics/ContextVariables/PROD/context.csv.*

→ Seleccione el check box **Include Header**.

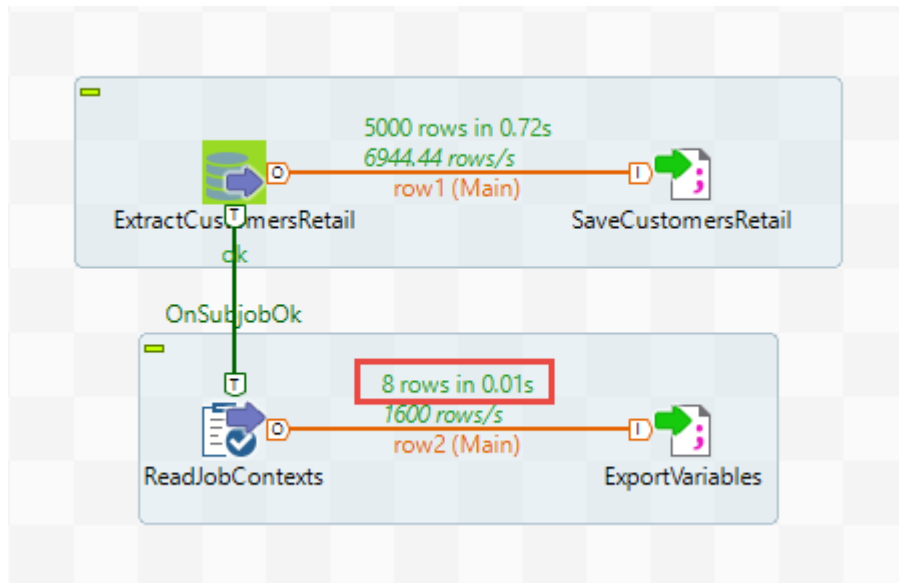


4.- Ejecutar el Job usando el contexto **Production**.

a.- Seleccione el contexto **Production**.

b.- Presione **Run**. El Job debe finalizar en **[exit code=0]**.

c.- En **Designer**, observe que 8 rows hayan sido enviadas al archivo.



d.- Abra la pestaña **Contexts** y confirme que las 8 variables estén allí.

Job(loadVariables 0.1) Contexts(loadVariabl... x

	Name
1	FilePath
2	DBConnect (from repository context)
3	DBConnect Database
4	DBConnect Password
5	DBConnect Server
6	DBConnect Port
7	DBConnect AdditionalParams
8	DBConnect Login
9	DataVolume (from repository context)
10	Limit

e.- Busque en la carpeta el archivo **context.csv** para confirmar que se ha creado.

This PC > Local Disk (C:) > StudentFiles > DIBasics > ContextVariables > PROD

Name	Date modified	Type	Size
context	3/10/2021 10:48 AM	CSV File	1 KB
CustomersRetail	3/10/2021 10:48 AM	CSV File	683 KB

## Carga de variables desde un archivo

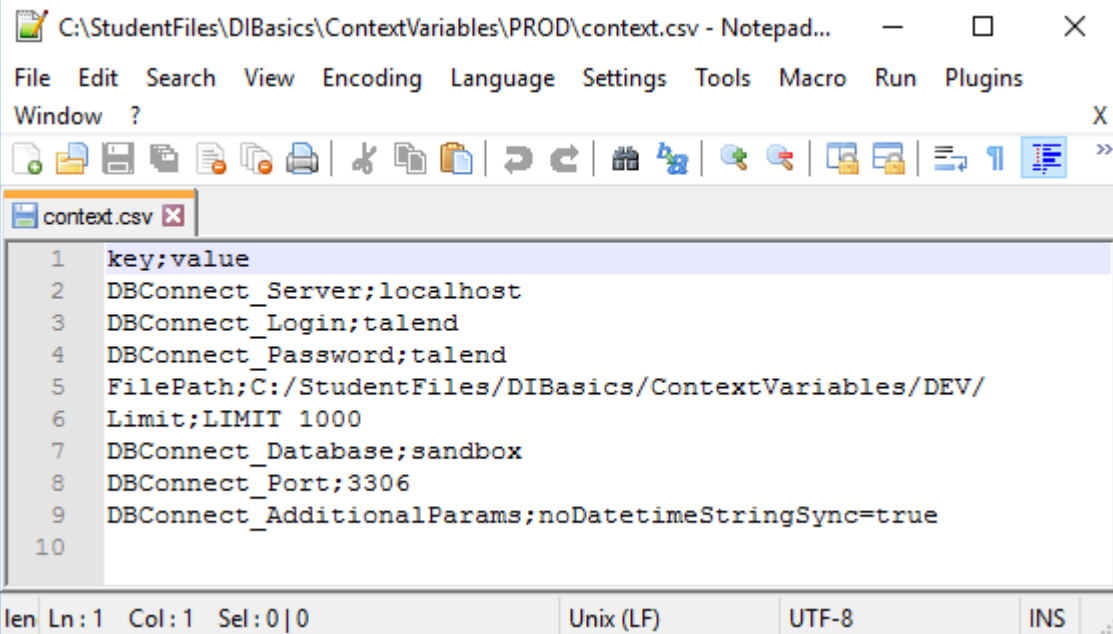
Ahora que ha exportado variables a un archivo, conoce la estructura del archivo para ir un paso más allá: cargar variables desde un archivo.

En este ejercicio, modificará un valor de variable en el archivo y cargará su contenido en el contexto del Job. En la ejecución, para el contexto seleccionado, los valores de las variables cargados desde el archivo sobrescriben los valores de las variables de contexto.

1.- Actualice el archivo.

Edite el archivo **context.csv** y actualice el server de la base de datos de **localhost** a **127.0.0.1**.

a.- Haga click derecho en **context.csv** y seleccione **Edit with Notepad++**.



The screenshot shows the Notepad++ application window titled 'C:\StudentFiles\DIBasics\ContextVariables\PROD\context.csv - Notepad...'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, and Plugins. The toolbar contains various icons for file operations and editing. The status bar at the bottom indicates 'Ln: 1 Col: 1 Sel: 0 | 0', 'Unix (LF)', 'UTF-8', and 'INS'. The main text area displays the following content:

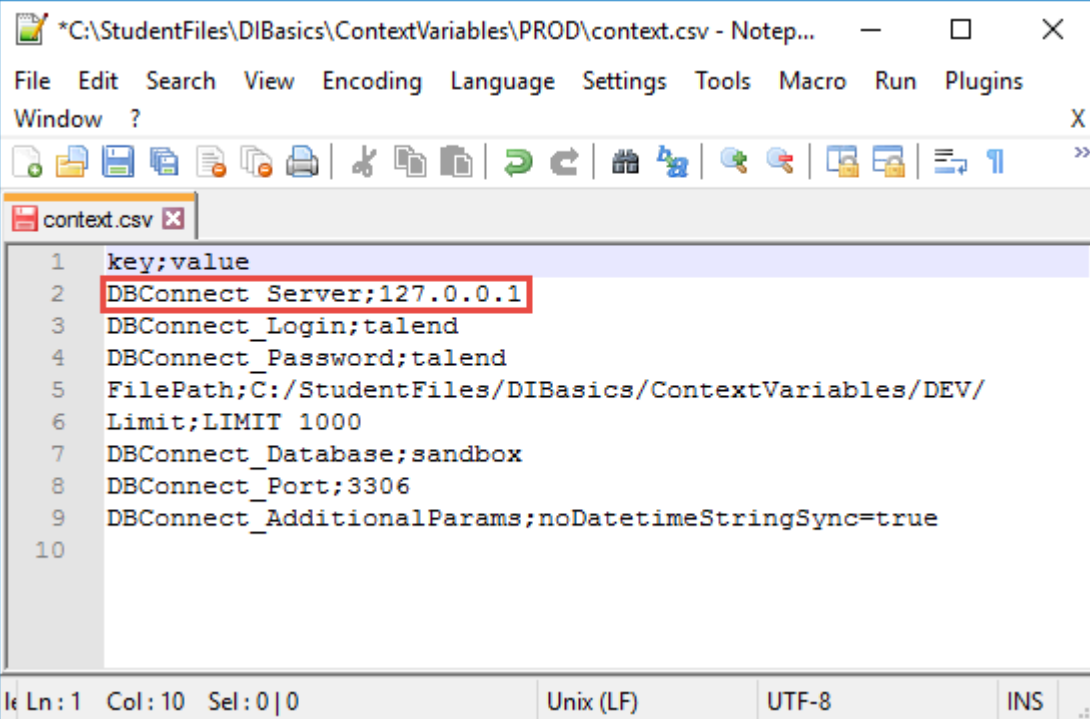
```

1 key;value
2 DBConnect_Server;localhost
3 DBConnect_Login;talend
4 DBConnect_Password;talend
5 FilePath;C:/StudentFiles/DIBasics/ContextVariables/DEV/
6 Limit;LIMIT 1000
7 DBConnect_Database;sandbox
8 DBConnect_Port;3306
9 DBConnect_AdditionalParams;noDatetimeStringSync=true
10

```

Las variables se han exportado en dos columnas: una para la clave y otra para el valor.

b.- A la derecha de la clave **DBConnect\_Server**, reemplace **localhost** por la IP **127.0.0.1**.



```

1 key;value
2 DBConnect_Server;127.0.0.1
3 DBConnect_Login;talend
4 DBConnect_Password;talend
5 FilePath;C:/StudentFiles/DIBasics/ContextVariables/DEV/
6 Limit;LIMIT 1000
7 DBConnect_Database;sandbox
8 DBConnect_Port;3306
9 DBConnect_AdditionalParams;noDatetimeStringSync=true
10

```

c.- Guarde las modificaciones y cierre el editor.

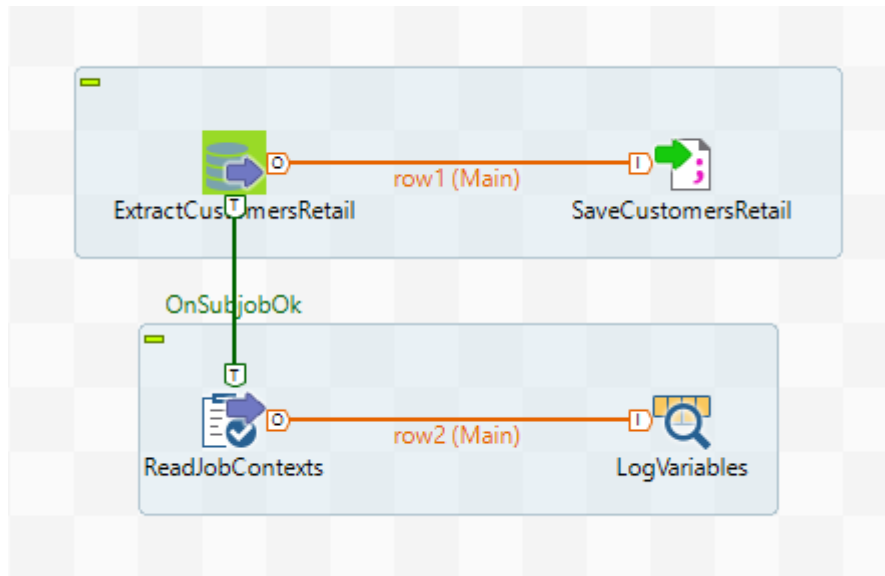
2.- Elimine el archivo de salida.

Para ver las variables en la consola cuando se ejecute, reemplace el último componente tFileOutputDelimited con un componente tLogRow.

a.- En **Designer**, seleccione el componente **ExportVariables** y presione **Delete**.

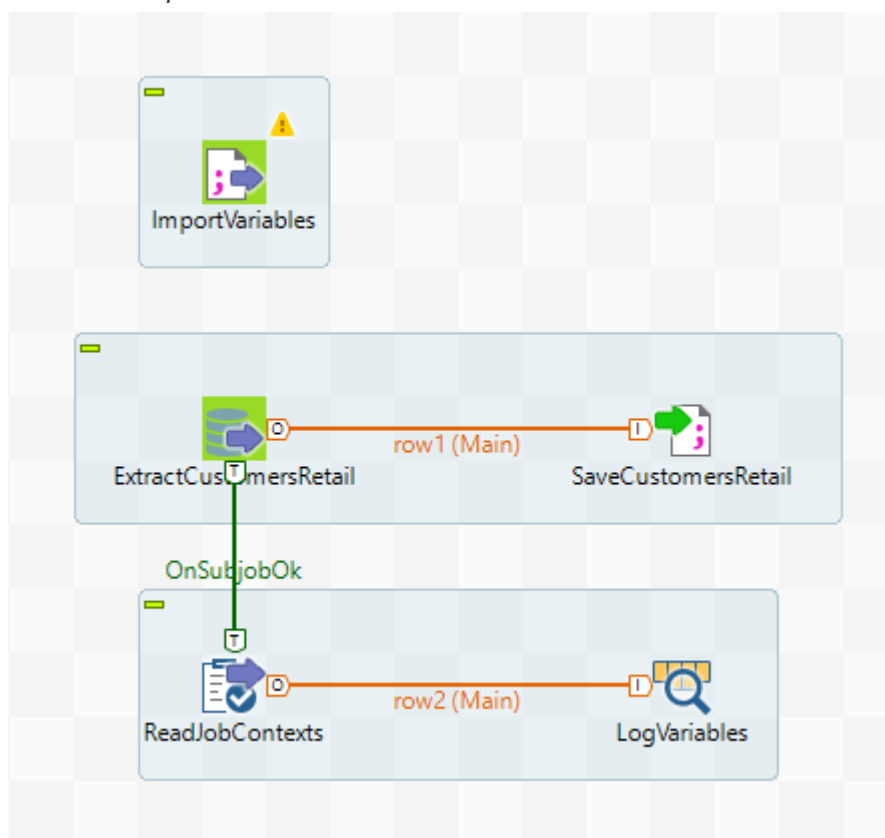
b.- A la derecha de **tContextDump**, coloque un componente **tLogRow** y nómbrelo *LogVariables*.

c.- Conecte **tContextDump** a **tLogRow** usando **Main** row.



3.- Agregue un componente **tFileInputDelimited** para cargar el contenido del archivo **context.csv**.

a.- Coloque un componente **tFileInputDelimited** sobre el componente **tDBInput** y nómbrelo **ImportVariables**.



b.- Configure el componente **tFileInputDelimited** con estos parámetros:

→ Para **File name**, haga click en [...] y seleccione la ruta **C:/StudentFiles/DIBasics/ContextVariables/PROD/context.csv**.

→ Para editar el schema manualmente, a la derecha de **Edit schema**, presione [...].

Cree dos columnas **String** y nómbrelas *key* y *value*.

Column	Key	Type	N.	Date Patte...	Len...	Prec...	De...	Co...
key	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>					
value	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>					

→ Para **Header**, ingrese 1.

Property Type: Built-In

Schema: Built-In Edit schema

File name/Stream: C:/StudentFiles/DIBasics/ContextVariables/PROD/context.csv

Row Separator: \n Field Separator: ,

Header: 1 Footer: 0 Limit:

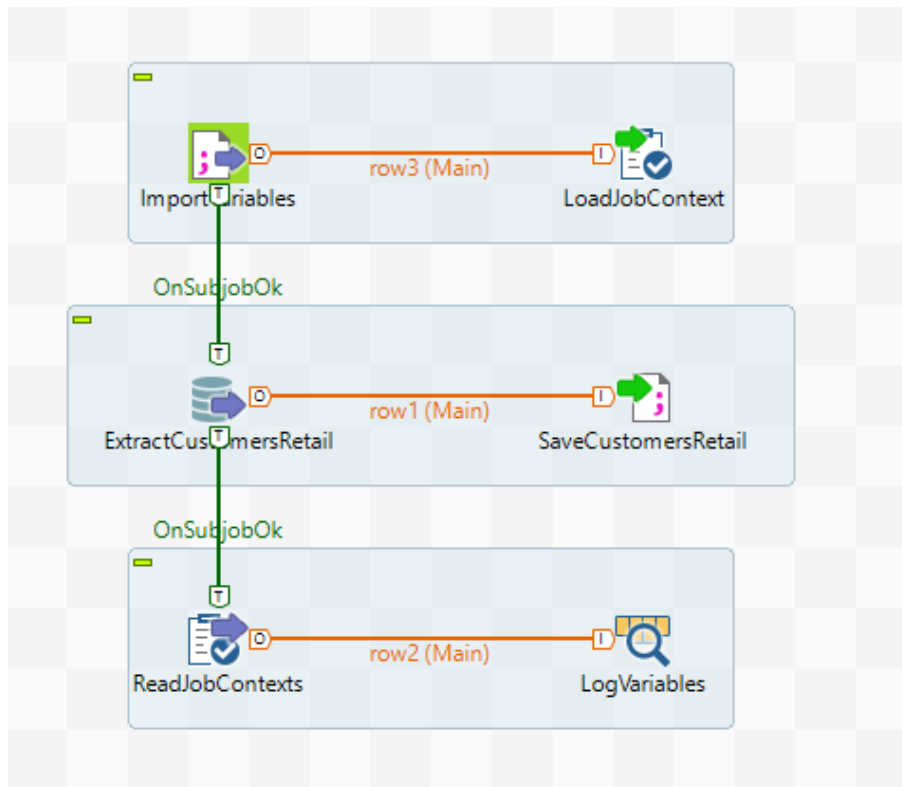
☒ Skip empty rows ☐ Uncompress as zip file ☐ Die on error

**4.-** Agregue un componente **tContextLoad** para actualizar el contexto usando los valores canalizados por el componente **tFileInputDelimited**.

**a.-** A la derecha del componente **tFileInputDelimited**, coloque un componente **tContextLoad** y nómbrelo *LoadJobContext*.



b.- Conecte los dos componentes usando **Main** row, luego conecte **tFileInputDelimited** a **tDBInput** usando un trigger **On Subjob Ok**.



5.- Seleccione el contexto **Production** y ejecute el Job.

a.- En la pestaña **Run**. Seleccione el contexto **Production**.

b.- Presione **Run**. Las variables se mostrarán por consola.

**Job loadVariables**

Basic Run  
Debug Run  
Advanced settings  
Target Exec  
Memory Run

Execution

Run Kill Clear

```

Starting job loadVariables at 08:26 14/12/2018.

[statistics] connecting to socket on port 3728
[statistics] connected
DBConnect_Server|127.0.0.1
DBConnect_Login|talend
DBConnect_Password|talend
FilePath|C:/StudentFiles/DIBasics/ContextVariables/DEV/
Limit|LIMIT 1000
DBConnect_Database|sandbox
DBConnect_Port|3306
DBConnect_AdditionalParams|noDatetimeStringSync=true
[statistics] disconnected

Job loadVariables ended at 08:26 14/12/2018. [exit code=0]
                
```

Line limit 100 Wrap

Production

Name
FilePath
Limit
DBConnect_Additi...
DBConnect_Datab...
DBConnect_Login
DBConnect_Passw...
DBConnect_Port
DBConnect_Server

Confirme que el server de la base de datos sea **127.0.0.1**. Esto no afectará al Job, el cual debe finalizar con **[exit code=0]**.

## Resumen

---

En este módulo aprendiste cómo:

- Crear variables de contexto built-in específicas para un Job
- Usar variables de contexto en la configuración de los componentes de Talend
- Ejecutar un Job usando variables de contexto
- Conectar un Job con una base de datos usando variables de contexto
- Crear variables de contexto en un repositorio que podrán usar todos los Jobs
- Cargar variables de contexto desde un flujo