

# AYI ACADEMY: TALEND

## Material práctico



### Módulo 7: Java family

## Objetivos

---

Después de completar este módulo, ud podrá:

- Diferenciar en la paleta de componetes los diferentes componentes de la familia Java.
- Ejecutar una simple instrucción java usando el componente tJava
- Manipular flujos de datos usando los componente tJavaRow y tJavaFlex
- Crear Routines Java y llamarlas desde los componentes de Talend

## Caso de Uso

---

A veces es necesario procesar datos muy específicos del negocio, o demasiados complejos para ser transformados por uno de los componentes disponibles en la paleta de Talend. Para estos casos, podemos usar código Java custom.

En Talend Studio, podemos usar código java en diferentes lugares:

- En los campos editables de los componentes.
- En los componentes Java específicos: tJava, tJavaRow, y tJavaFlex
- En las routines Java

Cada uno de estos métodos opera diferente. En este módulo vamos a enseñar como y cuando usarlos.

## Ejercicio

---

Tienes un archivo de origen que contiene filas de datos. Tu trabajo es generar una identificación única (UID) por cada uno de los clientes, usando algunas reglas de generación de ID. Puedes usar código Java custom en tu Talend Job para implementar estas reglas.

## Tarea

---

Antes de desarrollar nuevos Jobs en Talend Studio, inicia el Talend Studio y abre un proyecto en tu workspace.

## Abriendo el proyecto.

---

1. Inicie tu Talend.



2. Abra el proyecto donde viene trabajando: DIBASICS..
3. Configure la perspectiva de integración.  
En la barra de herramientas arriba del área de trabajo, asegúrese que la perspectiva

de Integración esté seleccionada.



## Configurando

---

La primera y más sencilla forma de usar el código Java en Talend es escribir instrucciones simples en los componentes existentes. Es posible que ya lo esté haciendo sin siquiera darse cuenta: llamando a las variables de contexto, usando una rutina o una variable de componente, definiendo un enlace "if", todo esto en realidad está usando instrucciones Java.

En este ejercicio, tu tarea es crear un Identificador de Usuario (UID) para una lista de clientes.

1. Abra Talend Studio, seleccione la perspectiva de **Integration**, y cree una carpeta llamada *JavaFamily*.
2. Dentro de la carpeta *JavaFamily* cree un nuevo **Standard Job** llamado *useCustomJavaCode*.
3. Cree un **File delimited** metadata llamado *Customers*.
  - a. El archivo *Customers* se encuentra en *C:\Student Files\DIAdvanced\raw\_customers\_extract.csv*.
  - b. Configure su metadata con la siguiente información: el separator es una coma, y el archivo contiene un header line que contiene el nombre de las columnas.

Use el screenshot de abajo si necesita ayuda.

#### File - Step 3 of 3

Update an existing Metadata File on repository  
Define the setting of the parse job



File Settings

Encoding

UTF-8

Field Separator

Comma

Corresponding Character

" "

Row Separator

Standard EOL

Corresponding Character

"\n"

Escape Char Settings

☐ CSV
 ☒ Delimited

Escape Char

Empty

Text Enclosure

Empty

☐ Split row before field

Rows To Skip

If any rows must be ignored, specify the following parameters

Header

☒

1

Footer

☐

☐ Skip empty row

Limit Of Rows

If the number of lines must be limited, specify this number

Limit

☐

Preview

Output

☒ Set heading row as column names
 

Refresh Preview

Firstname	Lastname	Address	City	Zipcode	State	Country	
Carole	Fortner	3739 Buena Vista Avenue	North Bend	97459	Oregon	USA	
Terri	Lockhart	890 Edgewood Road	Pine Bluff	71601	Arizona	USA	
Eugene	Moore	1964 Rebecca Street	Oak Brook	60523	Illinois	USA	
James	Rew	2662 Clousson ROad	Mobile	51039	Iowa	USA	
John	Warren	1122 Walt Nuzum Farm Road	Brighton	14623	New York	USA	
Teresa	Jones	4469 Black Stallion Road	Lexington	40511	Kentucky	USA	
Karolin	Nacht	Koepenicker Str. 86	Dachsenhausen	56340		Deutschland	

Export as context

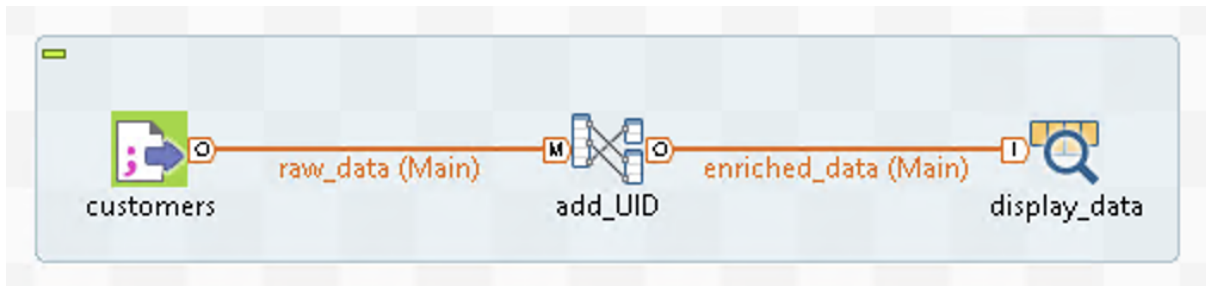
Revert Context

- c. Configurar el schema deberia ser sencillo: todas las columnas son string, y los nombres son provistos por el header. Podría precisar ayuda, use como referencia el screenshot de abajo. Click en el botón **Finish** cuando complete la configuración.

Description of the Schema								
Column	Key	Type	<input checked="" type="checkbox"/> N..	Date Pattern (Ctrl+Sp...	Length	Precision	Default	Comment
Firstname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9	0		
Lastname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		12	0		
Address	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		26	0		
City	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		18	0		
Zipcode	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9	0		
State	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9	0		
Country	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0		

## Usando Java en los componentes standars

En este Job, use un componente tMap para agregar una nueva columna identifier en la informacion del cliente. El identificador será generado usando una simple línea de código Java dentro del componente tMap. Así se verá finalmente su Job:



1. Drag and drop **customers metadata** que creó recién, dentro del área de diseño de su Job. Como queremos que lea los datos desde el archivo, elija usar el componente **tFileInputDelimited** cuando le pregunten.
2. Agregar un componente nuevo de **tMap** a la derecha del componente **customers**, liguelo con **Main** row, y renombre el tMap a **add\_UID**. Como buena practica, renombre el Main row como **raw\_data**.
3. Double-click en el componente **tMap** para abrirlo y configurarlo:
  - a. Cree un nuevo output llamado **enriched\_data**.
  - b. Agregue todas las columnas desde **raw\_data** input en **enriched\_data** output.
  - c. Agregue una nueva columna en el output, nombrela **ID**, seteela como de **Type String**, y nuevala como primera fila del schema. Su output schema deberia verse como la imagen de abajo:

enriched\_data

Column	Key	Type	<input checked="" type="checkbox"/> N.	Date Pattern (Ctrl+S...	Length
ID	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		
Firstname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9
Lastname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		12
Address	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		26
City	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		18
Zipcode	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9
State	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		9
Country	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15

4. Agregar código Java en el **Expression** field de la columna **ID**.  
 Como ya sabrá, es posible utilizar código Java en casi todos los campos editables de Talend Studio. Si presta atención al campo Expresión en la interfaz de tMap, se dará cuenta de que ya tiene algo de Java básico: **raw\_data.Firstname**, por ejemplo, no es nada más que variable de Java que apunta al dato **Firstname** en el flujo **raw\_data**.  
 Ingrese la siguiente línea de código Java en el **Expression** field de la columna **ID**:  

```
String.format("%10s",  
Numeric.sequence("s1",1,1)+"").replace(' ', '0');
```

 Esta expresión de Java utiliza una rutina numérica que incrementa una secuencia cada vez que se llama, luego formatea el número de secuencia en una cadena de 10 caracteres, reemplazando cualquier carácter no utilizado por ceros.  
 Click en el botón **Ok** button cuando termine.

- Finalice el Job agregando un componente **tLogRow** a la derecha del componente **tMap**. Ligue **enriched\_data** output con el componente **tLogRow** y ejecute el Job.

El resultado muestra la nueva columna de ID incremental junto con los datos del cliente. El resultado muestra la nueva columna de ID incremental junto con los datos del cliente

```
[statistics] connecting to socket on port 3629
[statistics] connected
0000000001|Carole|Fortner|3739 Buena Vista Avenue|North Bend|97459|Oregon|USA
0000000002|Terri|Lockhart|890 Edgewood Road|Pine Bluff|71601|Arizona|USA
0000000003|Eugene|Moore|1964 Rebecca Street|Oak Brook|60523|Illinois|USA
0000000004|James|Rew|2662 Clousson ROad|Mevile|51039|Iowa|USA
0000000005|John|Warren|1122 Walt Nuzum Farm Road|Brighton|14623|New York|USA
0000000006|Teresa|Jones|4469 Black Stallion Road|Lexington|40511|Kentucky|USA
0000000007|Karolin|Nacht|Koepenicker Str. 86|Dachsenhausen|56340||Deutschland
0000000008|Jessika|Fielder|Brandenburgische Str. 45|Carlsberg|67316||Deutschland
0000000009|Matthias|Eggers|Kieler Str. 32|Tittmoning|84529||Deutschland
0000000010|Jörg|Grunewald|Unter den Linden 75|Magdeburg|39014||Deutschland
```

## Componente tJava

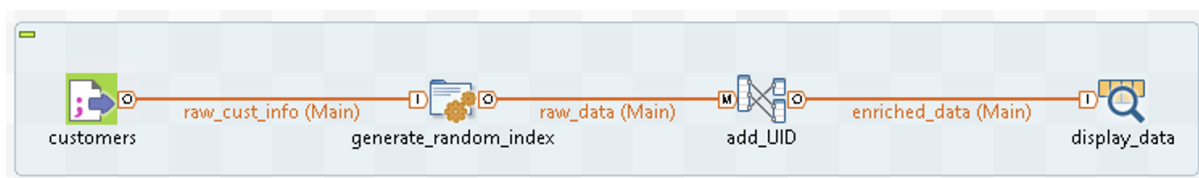
El componente tJava es el más simple componente Java de la paleta de Talend.

Puedes usarlo para correr una o más líneas de código Java con las siguientes restricciones:

- El código solo se ejecuta una vez; no se ejecuta en cada row de datos, sino una vez para todo el flujo.
- El componente no puede acceder a las rows de datos en modo de lectura o escritura, por lo que no es adecuado para ningún código de procesamiento de datos.e
- El componente puede acceder a las variables globales y de contexto.

## Agregando instrucciones con el componente tJava

Este trabajo se basa en el anterior. En el componente tJava, genera un número de lote que se usa más adelante en el componente tMap para crear la identificación del cliente. Así es como debería verse el trabajo final:



- Duplica el Job **useCustomJavaCode** y renombra la copia como **useTJava**. abre el Job **useTJava**.

2. Agrega el componente **tJava** entre el componente **customers tFileInput** y el componente de **tMap**. Nómbralo como *generate\_random\_index*, y, como muestra el screenshot de arriba, ligalo con los componentes.

3. Crea una variable de contexto *UID* de tipo **String**. Almacena el valor del número de índice generado en el componente tJava.

4. en la vista del componente tJava , en **Basic settings**, setear en **Code** la siguiente línea:

```
context.UID = Numeric.random(1,100)+"";
```

Esta línea de código genera un número aleatorio entre 1 y 100, lo concatena a una cadena vacía para convertir el tipo numérico en cadena y almacena el resultado en la variable de contexto UID.

En casos más complejos, puede escribir más de una línea de código. También es posible importar librerías Java en el panel **Advanced settings**.

5. Double-click en el componente **tMap** para abrirlo para configurar. Modifique la Expression de la columna ID column usando el siguiente código:

```
context.UID+String.format("%10s",  
Numeric.sequence("s1",1,1)+"").replace(' ', '0');
```

Esto concatena el índice almacenado en UID con la secuencia que ya conoce, agregando así un número de lote su cliente.

6. Ejecute the Job.

La identificación del cliente ahora debe estar compuesta por el índice aleatorio más el número de secuencia.

```
[statistics] connecting to socket on port 3939  
[statistics] connected  
730000000001|Carole| Fortner| 3739 Buena Vista Avenue| North Bend| 97459| Oregon| USA  
730000000002|Terri| Lockhart| 890 Edgewood Road| Pine Bluff| 71601| Arizona| USA  
730000000003|Eugene| Moore| 1964 Rebecca Street| Oak Brook| 60523| Illinois| USA  
730000000004|James| Rew| 2662 Clousson ROad| Merville| 51039| Iowa| USA  
730000000005|John| Warren| 1122 Walt Nuzum Farm Road| Brighton| 14623| New York| USA  
730000000006|Teresa| Jones| 4469 Black Stallion Road| Lexington| 40511| Kentucky| USA  
730000000007|Karolin| Nacht| Koepenicker Str. 86| Dachsenhausen| 56340|| Deutschland  
730000000008|Jessika| Fielder| Brandenburgische Str. 45| Carlsberg| 67316|| Deutschland  
730000000009|Matthias| Eggers| Kieler Str. 32| Tittmoning| 84529|| Deutschland  
730000000010|Jörg| Grunewald| Unter den Linden 75| Magdeburg| 39014|| Deutschland  
730000000011|Demi| Sutton| 53 Telford Street| Began| CF3 7FY|| United Kingdom  
730000000012|Lily| George| 74 Guildford Rd| East Lutton| YO17 7QD|| United Kingdom
```

Tenga en cuenta que el índice generado aleatoriamente es el mismo para todas los rows de datos porque el código en un componente tJava se ejecuta solo una vez cuando pasa un flujo de datos a través de él. Por el contrario, el número de secuencia se incrementa para cada fila, ya que el componente tMap ejecuta todas las expresiones cada vez que procesa un row del flujo de datos..

## Diferenciando entre los componentes tJava

A diferencia del componente **tJava**, **tJavaRow** y **tJavaFlex**:

- Pueden acceder al flujo de los datos

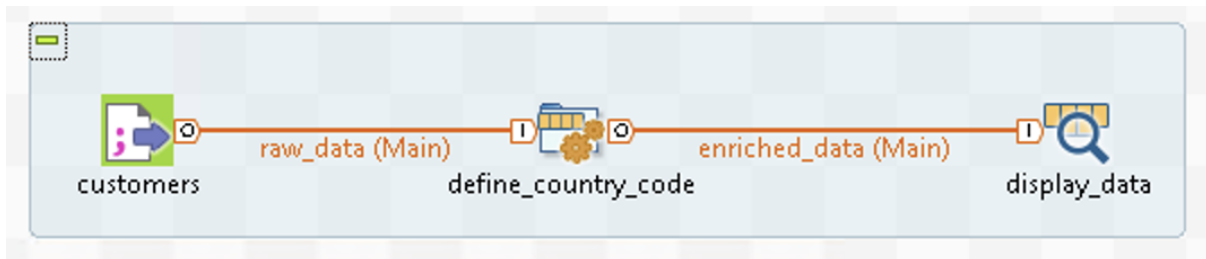
- Se ejecutan tantas veces como rows de datos pasan a través de ellos (si procesa un flujo de datos con 600 rows, el código dentro del componente también se ejecutará 600 veces, una vez por cada row)

Las principales diferencias:

- **tJavaRow** tiene un único bloque de código y no propaga datos automáticamente
- **tJavaFlex** tiene tres bloques de código (Start code, Main code, y End code) y propaga los datos automáticamente

## Usando el componente tJavaRow

En este ejercicio, agregará un poco más de complejidad a la creación del ID de cliente. Según el valor de la columna país, se agrega un código de país al comienzo de la identificación. El trabajo final debería verse así:



1. Crear un nuevo **Standard Job** y nombrelo *useTJavaRow*.
2. Arraste el archivo **customers** de metadata como un componente **tFileInputDelimited** en el área de diseño.
3. Pegue un componente **tJavaRow** desde la paleta de componentes y nombrelo *define\_country\_code*.
4. Ligue los componentes **customers** **tFileInputDelimited** con el componente **define\_country\_code** **tJavaRow** con un **Main** row. Rename the **Main** row como *raw\_data*.
5. En la vista del componente **tJavaRow**, click **Edit schema**.  
Tenga en cuenta que el esquema de entrada de los datos del cliente no se propaga automáticamente al esquema de salida. Es parte de la configuración de este componente elegir qué columnas de datos se enviarán.
6. Propagate the whole schema by either clicking the double arrow in the **Edit schema** wizard or by clicking the **Sync columns** button on the **Component** view. Propagate todo el esquema haciendo doble-clic en la flecha doble del asistente Editar esquema o haciendo clic en el botón Sincronizar columnas en la vista de componentes.



7. Actualice el esquema de salida agregando una columna final denominada ID con el tipo String. El esquema de salida debería verse así:

Column	Key	Type	<input checked="" type="checkbox"/> N..	Date Pa...	Len...	Pre...
Firstname	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
Lastname	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		12	0
Address	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		26	0
City	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		18	0
Zipcode	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
State	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
Country	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		15	0
ID	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			

8. En la vista del componente, click **Generate code**. Esto desencadena la generación del código que propaga los datos desde la fila de entrada a la fila de salida. Como se mencionó en la introducción de esta sección, el componente tJavaRow no propaga datos automáticamente. Si una fila de salida no recibe ninguna instrucción explícita en el código, aparece vacía o incluso nula, lo que puede provocar errores.
9. Puedes enriquecer la caja **Code** de la vista de **Component** agregando tus propias líneas de código. Puedes encontrar código en la sección *C:\StudentFiles\DIAdvanced\tJavaRow\_code.txt*. Copia el contenido del archivo y pégalo en la caja **Code**, asegurándose de que reemplaza todo el código Java existente.

```
String countryCode = "";
switch (input_row.Country.trim().toLowerCase()) {
    case "usa":
        countryCode = "US";
        break;
    case "deutschland":
        countryCode = "DE";
        break;
    case "united kingdom":
        countryCode = "UK";
        break;
    case "france":
        countryCode = "FR";
        break;
    case "italia":
        countryCode = "IT";
        break;
}

//Code generated according to input schema and output schema
output_row.Firstname = input_row.Firstname;
output_row.Lastname = input_row.Lastname;
output_row.Address = input_row.Address;
output_row.City = input_row.City;
output_row.Zipcode = input_row.Zipcode;
output_row.State = input_row.State;
output_row.Country = input_row.Country;
output_row.ID = countryCode+String.format("%10s", Numeric.sequence("s1",1,1)+").replace(' ', '0'); ;
```

Este código Java lee el valor de la columna País para cada row de entrada y genera el código de país correspondiente mediante una instrucción de cambio. Luego, propaga todos los datos y completa la ID de la columna de salida con el código de país concatenado con un número de secuencia..

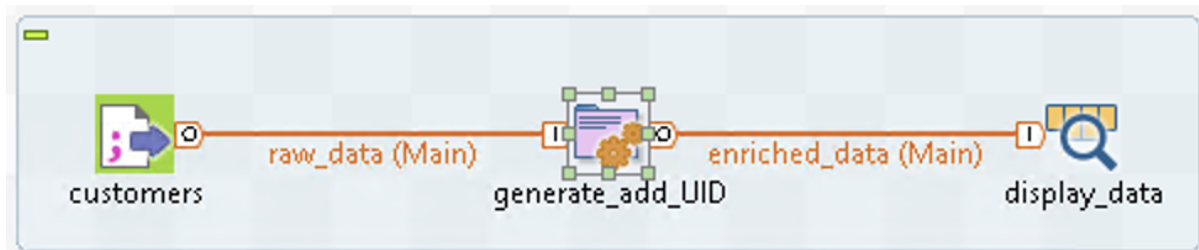
10. Agregar un componente **tLogRow** a tu Job, y ligalo al componente **tJavaRow**.
11. Ejecuta el Job y q and asegúrese de que sus identificaciones sean consistentes con cada país del cliente.

```
[statistics] connecting to socket on port 3862
[statistics] connected
Carole| Fortner| 3739 Buena Vista Avenue| North Bend| 97459| Oregon| USA|US0000000001
Terri| Lockhart| 890 Edgewood Road| Pine Bluff| 71601| Arizona| USA|US0000000002
Eugene| Moore| 1964 Rebecca Street| Oak Brook| 60523| Illinois| USA|US0000000003
James| Rew| 2662 Clousson ROad| Movile| 51039| Iowa| USA|US0000000004
John| Warren| 1122 Walt Nuzum Farm Road| Brighton| 14623| New York| USA|US0000000005
Teresa| Jones| 4469 Black Stallion Road| Lexington| 40511| Kentucky| USA|US0000000006
Karolin| Nacht| Koepenicker Str. 86| Dachsenhausen| 56340|| Deutschland|DE0000000007
Jessika| Fielder| Brandenburgische Str. 45| Carlsberg| 67316|| Deutschland|DE0000000008
Matthias| Eggers| Kieler Str. 32| Tittmoning| 84529|| Deutschland|DE0000000009
Jörg| Grunewald| Unter den Linden 75| Magdeburg| 39014|| Deutschland|DE0000000010
Demi| Sutton| 53 Telford Street| Began| CF3 7FY|| United Kingdom|UK0000000011
Lily| George| 74 Guildford Rd| East Luton| YO17 7QD|| United Kingdom|UK0000000012
Evie| Coates| 48 Bootham Crescent| Rivar| SN8 3WL|| United Kingdom|UK0000000013
Tyler| Morrison| 3 Gloddaeth Street| Birdlip| GL4 2PE|| United Kingdom|UK0000000014
Cameron| Dale| 26 Mounthoolie Lane| Summercourt| TR8 6XR|| United Kingdom|UK0000000015
Nicolas| Rhéaume| 34 rue de la Boétie| Pontault-Combault| 77340|| France|FR0000000016
Philippe| Cadieux| 23 rue des Lacs| Hénin-Beaumont| 62110|| France|FR0000000017
Julien| Déziel| 67 boulevard de Prague| Niort| 79000|| France|FR0000000018
```

## Usando el componente tJavaFlex

El trabajo anterior le permitió agregar instrucciones más largas y usar algunos datos en el flujo para enriquecer la creación de la identificación del cliente. Ahora el ID de cliente tiene un código de país que refleja el país del cliente. Pero el número de secuencia se comparte entre todos los clientes. Gracias a los diferentes bloques de código en el componente tJavaFlex, puede refinar el proceso inicializando una secuencia por país e incrementándolos cuando sea necesario, para que la secuencia de ID de cliente para cada país sea continua..

Su Job final debería verse así:



1. Cree un nuevo **Standard Job** y pongale por nombre *useTJavaFlex*.
2. Arrastre el archivo **customers** desde metadata como un componente **tFileInputDelimited** en el area de diseño.
3. Pegue un componente **tJavaFlex** desde la paleta de componentes y nombrello *generate\_add\_UID*.
4. Ligue los componentes **customers** **tFileInputDelimited** con **generate\_add\_UID** **tJavaFlex** mediante un **Main** row. Renombre el **Main** row como *raw\_data*.
5. En la vista **Component** del compoente **tJavaFlex**, haga click en **Edit schema**. Tenga en cuenta que el schema de entrada de los datos del cliente no se propaga automáticamente al esquema de salida. Es parte de la configuración de este componente elegir qué columnas de datos se enviarán.

6. Propagate the whole schema by either clicking the double arrow in the **Edit schema** wizard or by clicking the **Sync columns** button on the **Component** view. Propague todo el esquema haciendo doble clic en la flecha doble del asistente Editar esquema o haciendo clic en el botón Sincronizar columnas en la vista de componentes..
7. Actualice el schema de salida agregando al final una columna que nombrará *ID*, configurela de tipo **String**. Su schema de salida deberi verse:

Column	Key	Type	<input checked="" type="checkbox"/> N..	Date Pa...	Len...	Pre...
Firstname	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
Lastname	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		12	0
Address	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		26	0
City	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		18	0
Zipcode	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
State	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		9	0
Country	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>		15	0
ID	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			

8. En la vista del componente, Observe el check box **Data Auto Propagate**. Cuando está marcado, el componente propagará automáticamente todos los datos del row de entrada al row de salida correspondiente. No se necesita ninguna acción en el código Java.
9. Para llenar los bloques de código, abra el archivo ubicado en *C:\StudentFiles\DIAdvanced\JavaFlex\_code.txt*. Copy the **Start** block and paste it in the **Start code** box of the component. Then, copy the **Main** block from the file and paste it in the **Main code** box of the component. Copie el Start block y péguelo en el cuadro Start Code del componente. Luego, copie el Main block del archivo y péguelo en el Main Code del componente  
Examine el código e intente comprender lo que hace. Aquí hay algunos consejos:
  - el **Start code** se ejecuta una sola vez, cuando se inicializa el componente. Generalmente se usa para crear e inicializar variables u objetos..
  - El **Main code** se ejecuta para cada fila de datos que pasa por el componente.
  - The **End code** se ejecuta una sola vez, después de que se haya procesado la última fila de datos. Se utiliza principalmente para destruir y finalizar objetos.
  - Para acceder a la columna de una fila, escriba `name_of_the_row.name_of_the_column`.
10. Como habrás adivinado, el código de inicio inicializa un contador por país, así como dos variables: un código de país y un ID actual. Para cada fila de datos, la instrucción de cambio intenta hacer coincidir los datos del país con una de las condiciones. Una vez en una instrucción de caso, la variable countryCode recibe el valor apropiado y currentID recibe el valor actual de la secuencia para este país. Luego se incrementa la secuencia. Luego, el código completa la variable ID con el

código de país y el número de secuencia almacenado en la variable ID actual.  
Agrega un componente **tLogRow** en tu Job, y ligalo al componente **tJavaFlex**.  
Renombra el Main row como *enriched\_data*.

Asegúrese de cambiar el nombre de la fila con precisión. El **Main code** asignará valores usando ese nombre exacto.

11. Ejecute el Job, y asegúrese de que los ID generados sean consistentes: código de país apropiado y una secuencia continua para cada país.

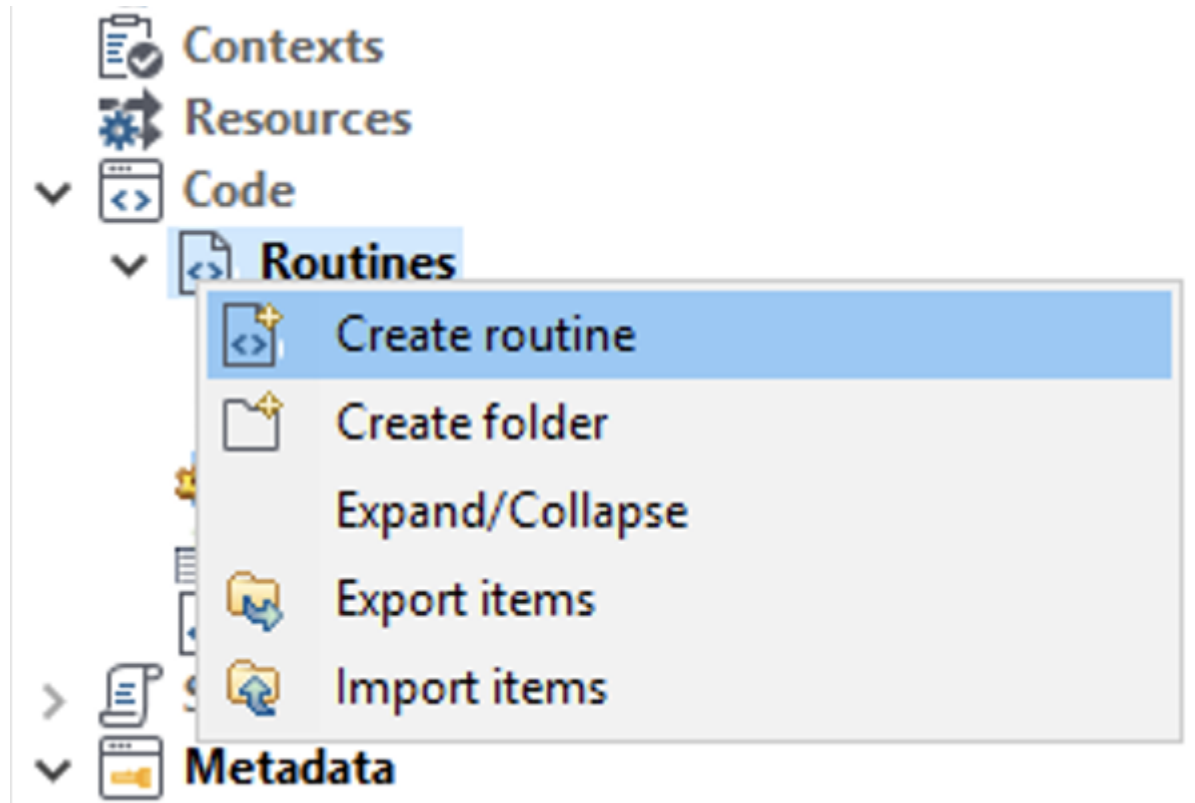
```
[statistics] connecting to socket on port 3701
[statistics] connected
Carole| Fortner| 3739 Buena Vista Avenue| North Bend| 97459| Oregon| USA|US0000000001
Terri| Lockhart| 890 Edgewood Road| Pine Bluff| 71601| Arizona| USA|US0000000002
Eugene| Moore| 1964 Rebecca Street| Oak Brook| 60523| Illinois| USA|US0000000003
James| Rew| 2662 Clousson ROad| Movile| 51039| Iowa| USA|US0000000004
John| Warren| 1122 Walt Nuzum Farm Road| Brighton| 14623| New York| USA|US0000000005
Teresa| Jones| 4469 Black Stallion Road| Lexington| 40511| Kentucky| USA|US0000000006
Karolin| Nacht| Koepenicker Str. 86| Dachsenhausen| 56340|| Deutschland|DE0000000001
Jessika| Fielder| Brandenburgische Str. 45| Carlsberg| 67316|| Deutschland|DE0000000002
Matthias| Eggers| Kieler Str. 32| Tittmoning| 84529|| Deutschland|DE0000000003
Jörg| Grunewald| Unter den Linden 75| Magdeburg| 39014|| Deutschland|DE0000000004
Demi| Sutton| 53 Telford Street| Began| CF3 7FY|| United Kingdom|UK0000000001
Lily| George| 74 Guildford Rd| East Lutton| YO17 7QD|| United Kingdom|UK0000000002
Evie| Coates| 48 Bootham Crescent| Rival| SN8 3WL|| United Kingdom|UK0000000003
Tyler| Morrison| 3 Gloddaeth Street| Birdlip| GL4 2PE|| United Kingdom|UK0000000004
Cameron| Dale| 26 Mounthoolie Lane| Summercourt| TR8 6XR|| United Kingdom|UK0000000005
Nicolas| Rhéaume| 34 rue de la Boétie| Pontault-Combault| 77340|| France|FR0000000001
Philippe| Cadieux| 23 rue des Lacs| Hénin-Beaumont| 62110|| France|FR0000000002
Julien| Déziel| 67 boulevard de Prague| Niort| 79000|| France|FR0000000003
Matilda| Guillemette| 15 rue Charles Corbeau| Evry| 91000|| France|FR0000000004
```

## Creating and using a routine

En este ejercicio, creará una rutina que funciona exactamente igual que el componente tJavaFlex de la sección anterior. Sin embargo, poner este código en una rutina te permite llamarlo en cualquier campo editable de un Job como método de clase, tantas veces como sea necesario.

1. Cree una nueva clase de routine.
  - a. En la vista de **Repository**, expanda **Code** y haga click-derecho sobre **Routines**.

- b. Elija **Create routine** desde el menu contextual.



- c. Nombre la nueva routine como *customRoutines* y haga click en **Finish**.  
Se abre una nueva ventana de código y muestra un esqueleto de código de clase.
- El código que necesitas esta en *C:\StudentFiles\DIAdvanced\customRoutines.txt*. Copie el contenido del archivo. Elimine todo el código del esqueleto de la clase y pegue el contenido del archivo en su lugar, luego guarde su archivo de clase. Eche un vistazo al código: aquí, los contadores se inicializan como variables estáticas, por lo que son accesibles para cualquier método de la clase. El resto del código está en un método estático llamado `generateUID` que toma solo un parámetro: el país del cliente.
  - Duplique el primer Job para este ejercicio, **useCustomJavaCode**, nombre la copia como *useRoutine*, y abra el Job *useRoutine*.
  - Abra el componente **tMap** . En lugar de la expresión simple que usó para completar la columna de ID, llame al método de clase `generateUID()`. En términos generales, para llamar a una rutina, use `class_name.method_name(argument1, argument2, ...)` .Recuerde que el nombre de las variables se basa parcialmente en el nombre de las **Rows**.
    - En **Expression** field de la columna **ID** , escriba `customRoutines.generateUID(raw_data.Country)` .
    - Click **Ok** para cerrar el tMap.



5. Ejecute el Job. Debería obtener el mismo resultado que con el componente tJavaFlex.

```
[statistics] connecting to socket on port 4026
[statistics] connected
US0000000001|Carole| Fortner| 3739 Buena Vista Avenue| North Bend| 97459| Oregon| USA
US0000000002|Terri| Lockhart| 890 Edgewood Road| Pine Bluff| 71601| Arizona| USA
US0000000003|Eugene| Moore| 1964 Rebecca Street| Oak Brook| 60523| Illinois| USA
US0000000004|James| Rew| 2662 Clousson ROad| Movile| 51039| Iowa| USA
US0000000005|John| Warren| 1122 Walt Nuzum Farm Road| Brighton| 14623| New York| USA
US0000000006|Teresa| Jones| 4469 Black Stallion Road| Lexington| 40511| Kentucky| USA
DE0000000001|Karolin| Nacht| Koepenicker Str. 86| Dachsenhausen| 56340|| Deutschland
DE0000000002|Jessika| Fielder| Brandenburgische Str. 45| Carlsberg| 67316|| Deutschland
DE0000000003|Matthias| Eggers| Kieler Str. 32| Tittmoning| 84529|| Deutschland
DE0000000004|Jörg| Grunewald| Unter den Linden 75| Magdeburg| 39014|| Deutschland
UK0000000001|Demij| Sutton| 53 Telford Street| Began| CF3 7FY|| United Kingdom
UK0000000002|Lily| George| 74 Guildford Rd| East Lutton| YO17 7QD|| United Kingdom
```

## Sumario

---

En este módulo aprendimos como:

- Diferenciar entre los componentes Java de la paleta de componentes
- Ejecutar una simple instrucción java usando el componente tJava
- Manipular rows de datos uando los componente tJavaRow y tJavaFlex
- Crear una routine Java y llamarla desde un componente estándar de Talend