JOB AID

Handling errors

Overview

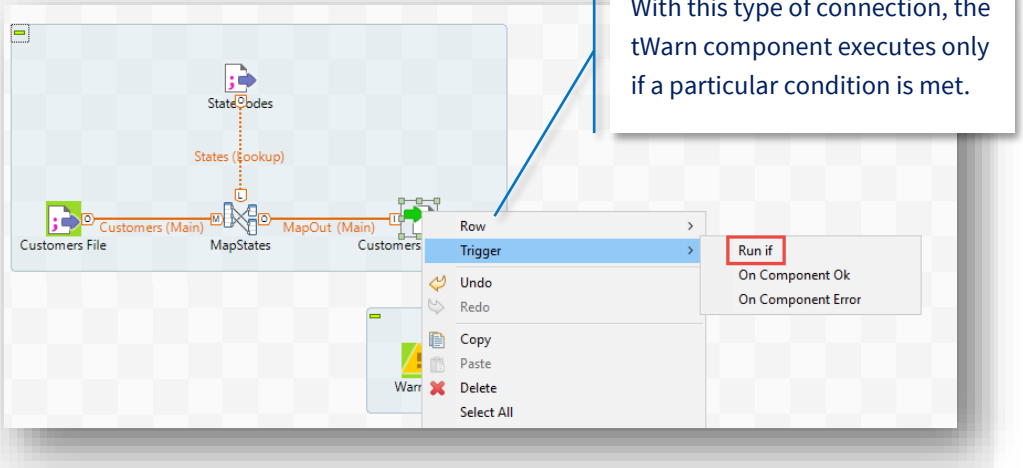
In a development or test environment, you can handle errors simply by observing the console. However, when your project is deployed to production, you want it to deal with errors automatically. In this module, you learned how to handle errors in a Job before publishing it to production. You also learned how to define error and warning messages.

Key steps

Follow these steps to raise a warning when an inner join rejects input data.

1 Use a tWarn component to raise a warning without killing a Job.

- Place a **tWarn** component in the **Designer**.
- Connect it to the component to the right of **tMap** using a **Run if** trigger.

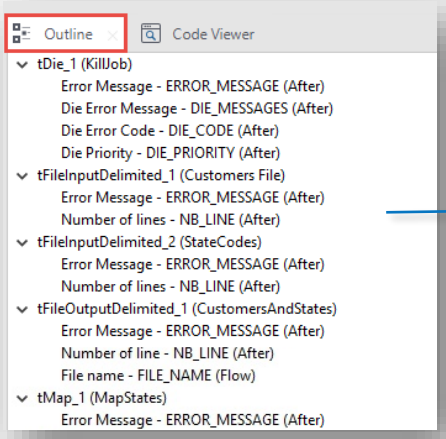


With this type of connection, the tWarn component executes only if a particular condition is met.

2 Before selecting the condition for a trigger, review the component variable you can use.

Each component has an associated list of predefined component variables. You can use them to keep track of component execution and configure other components in the Job.

- To display variables assigned to components in a Job, below the **Repository**, display the **Outline** view.



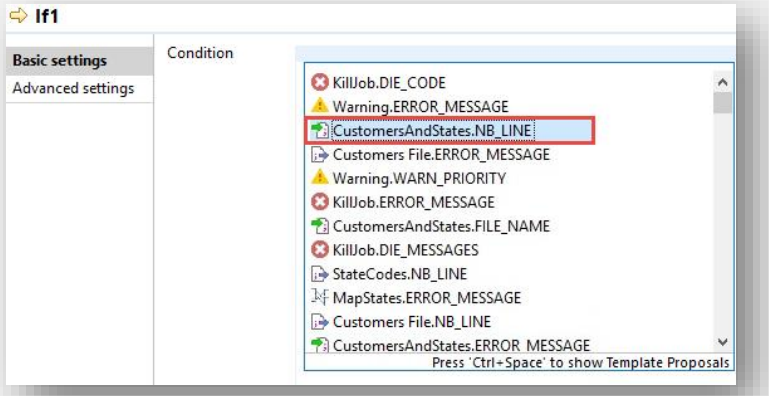
In this use case, to compare the number of rows before and after the **tMap** component, use the **NB_LINE** variables from **tFileInputDelimited_1 (Customers File)** and **tFileOutputDelimited_1 (CustomersAndStates)**.

JOB AID

3

Use component variables to select the trigger condition. Compare the number of rows before and after the **tMap** component.

1. In the **Designer**, select the connector and open the **Component** view.
2. In the **Condition** box, press **Ctrl+Space** to invoke an auto-completion list.
3. Double-click the first variable you want to use.
4. Complete the condition by entering a **less than** symbol (**<**) and the second variable.



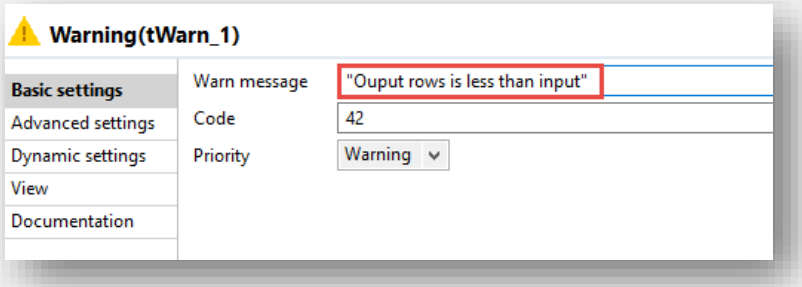
In this use case, the Condition field looks like this:

```
((Integer)globalMap.get("tFileOutputDelimited_1_NB_LINE"))<((Integer)globalMap.get("tFileInputDelimited_1_NB_LINE"))
```

4

Configure a warning message that will appear in the execution logs when the error occurs.

1. Double-click the **tWarn** component to open the **Component** view.
2. Update the **Warn message** text.

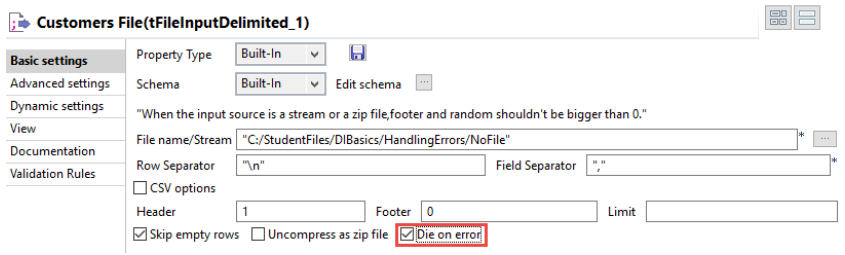


JOB AID

Tips

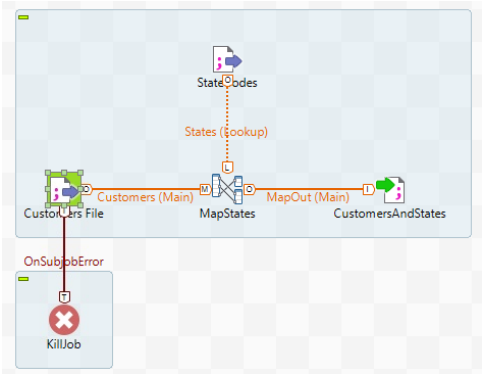
Other ways to manage component errors:

Die on error



- By default, a Job does not stop executing when a non-blocking component error occurs.
- You can use the **Die on error** option with many components.
- To stop a Job when a component generates an error, on the **Basic settings** tab in the **Component** view, select the check box for **Die on error**.

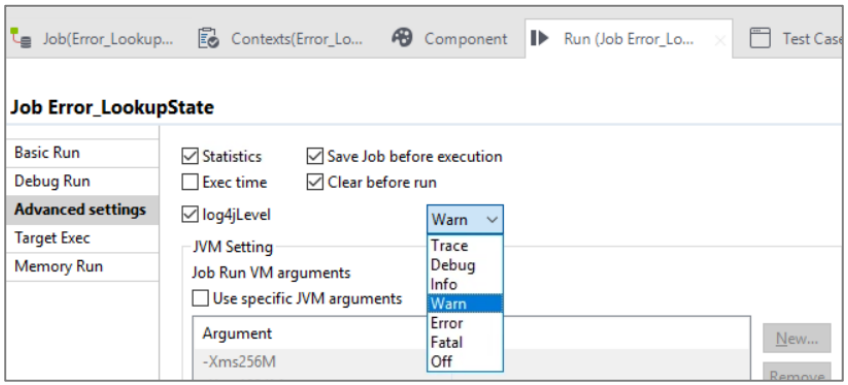
tDie



To kill a Job when a component or subJob generates an error, you can also use a **tDie** component.

1. Place a **tDie** component in the **Designer**.
2. Connect it to the component you want to monitor using an **On Component Error** or **On SubJob Error** trigger.
3. On the **Basic settings** tab in the **Component** view, you can specify a **Die message**.

Logging level



You can update the log level of a Job to control the messages that appear in the console.

1. Display the **Advanced settings** tab in the **Run** view.
2. Select the **log4jLevel** check box and choose a level on the list of values.

You can configure the **Log4jLevel** for all Jobs in a project in **File** > **Edit Project Properties**.